V-PITS : VIDEO BASED PHONOMICROSURGERY INSTRUMENT TRACKING SYSTEM

by

Ketan Surender

A thesis submitted in partial fulfillment of
the requirements for the degree of

Master of Science

(Electrical Engineering)

at the

UNIVERSITY OF WISCONSIN–MADISON

2011

To my parents and all the teachers I have ever had.

ACKNOWLEDGMENTS

This work could not have been done without the support of Dr. Jack Jiang. This project was done through The University of Wisconsin-Madison Laryngeal Physiology Lab of which he is the PI. I would also like to thank Adam, Ben, Carlos, Jason, Matt, and Zhixian from the lab for helping me with the project along the way. I would also like to thank Dr. Charles Dyer for providing guidance and comments on several aspects of the project. Finally, I would like to thank my adviser Dr. William Sethares who was very helpful throughout the entire project.

DISCARD THIS PAGE

TABLE OF CONTENTS

Page

Appendix

Page

DISCARD THIS PAGE

LIST OF TABLES

DISCARD THIS PAGE

LIST OF FIGURES

Figure Page

Appendix
Figure Page

Appendix
Figure Page

V-PITS : VIDEO BASED PHONOMICROSURGERY INSTRUMENT TRACKING SYSTEM

Ketan Surender

Under the supervision of Professor Dr. William Sethares

At the University of Wisconsin-Madison

V-PITS (Video Based Phonomicrosurgery Instrument Tracking System) is a system developed to estimate the trajectory of instruments performing a simulated phonomicrosurgery exercise. It first captures a synchronized pair of videos using a calibrated stereo camera rig. Next, it processes both videos with a tracking algorithm that is run off-line on each video twice, once for each instrument. The tracking algorithm detects a set of features in each frame related to instrument position and orientation. Finally, a trajectory estimation algorithm uses the feature data from both videos to estimate 3D trajectory.

V-PITS consists of a data acquisition and algorithm component that are described in this document. The algorithm component is based on a position estimation methodology which is also described. Experiments are reported that characterize system noise and displacement measurement accuracy. Uncertainty in the position reported by V-PITS due to noise was characterized as +/- 0.073 mm. Displacement measurement accuracy was characterized as 0.15 mm RMS.

Dr. William Sethares

ABSTRACT

V-PITS (Video Based Phonomicrosurgery Instrument Tracking System) is a system developed to estimate the trajectory of instruments performing a simulated phonomicrosurgery exercise. It first captures a synchronized pair of videos using a calibrated stereo camera rig. Next, it processes both videos with a tracking algorithm that is run off-line on each video twice, once for each instrument. The tracking algorithm detects a set of features in each frame related to instrument position and orientation. Finally, a trajectory estimation algorithm uses the feature data from both videos to estimate 3D trajectory.

V-PITS consists of a data acquisition and algorithm component that are described in this document. The algorithm component is based on a position estimation methodology which is also described. Experiments are reported that characterize system noise and displacement measurement accuracy. Uncertainty in the position reported by V-PITS due to noise was characterized as +/- 0.073 mm. Displacement measurement accuracy was characterized as 0.15 mm RMS.

Chapter 1

Introduction

1.1 Motivation

1.1.1 Technical Surgical Skill Evaluation

The success of surgery is attributable to the decision making ability and technical skill (dexterity) of a surgeon [5]. Through the use of classroom training and examinations, decision making skills can be taught and evaluated. Traditionally, technical surgical skill has been taught and evaluated in a less objective manner. In the traditional model, a surgical resident would develop technical skills by observing and performing surgeries under the supervision of senior faculty. This methodology is time consuming and many times does not contain standardized means for evaluating surgical skill [10]. Simulators and bench-top models have come into increasing use due to these issues. They provide for efficient training and structured evaluation of technical skill outside the operating room. By incorporating sensors into these simulated surgical setups, measurements of motion can be captured. Several quantitative methods for evaluating technical surgical skill have been developed based on processing this raw motion data .

1.1.2 Phonomicrosurgery Skill Evaluation

Phonomicrosurgery is a discipline within voice surgery. Typically, the purpose of this type of surgery is to improve the vibratory characteristics of the vocal folds [12]. Vibration of the vocal folds is a critical component in the production of regular speech. Figure 1.1 shows an

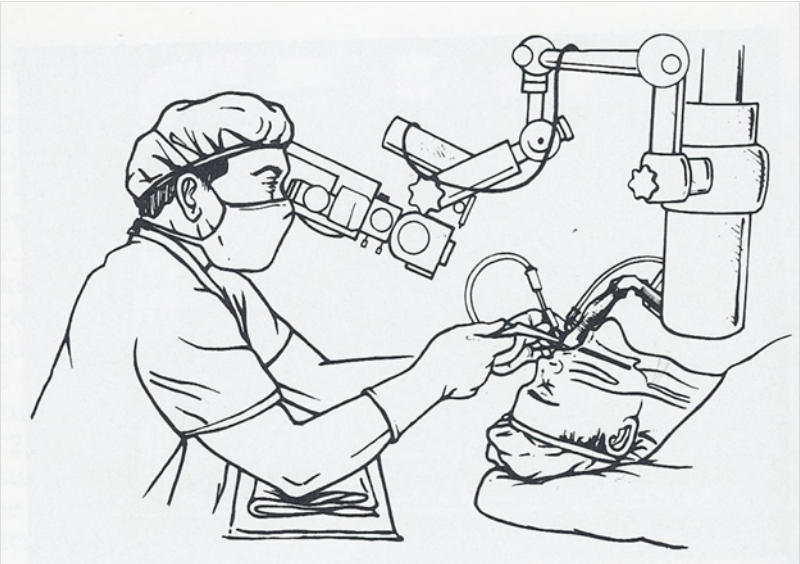

Figure 1.1 Illustration of Phonomicrosurgery [4].

illustration of a surgeon performing a phonomicrosurgery. A tube-like object called a laryngoscope is inserted into the patient's mouth during an operation. It gives the surgeon a direct view of the larynx within the neck. The vocal folds are part of the larynx. To operate on the vocal folds, surgical instruments are inserted into the laryngoscope. Figure 1.2 is an image of a forceps used for phonomicrosurgery. The surgeon views the tips of the instruments near the vocal folds through a surgical microscope.

A project had been initiated in the past related to the simulation of phonomicrosurgery. The goal of the project was to quantify technical skill and improvement using instrument motion data. Figure 1.3 is an image of a laryngeal dissection station (LDS). The bronze tube in the station is designed to simulate a laryngoscope. In the project, a paper target was mounted beyond the tip of the laryngoscope in the LDS. A subject performed a simulated exercise using forceps in his/her non-dominant hand and scissors in the dominant hand. The paper target would first be grasped using the forceps and then cut along a guideline using the scissors. This was repeated at multiple points on the target and was meant to simulate a vertical cutting technique used in phonomicrosurgery. A magnetic position sensing device was used to track the movements of the surgical instruments during this exercise. This device used a pair of

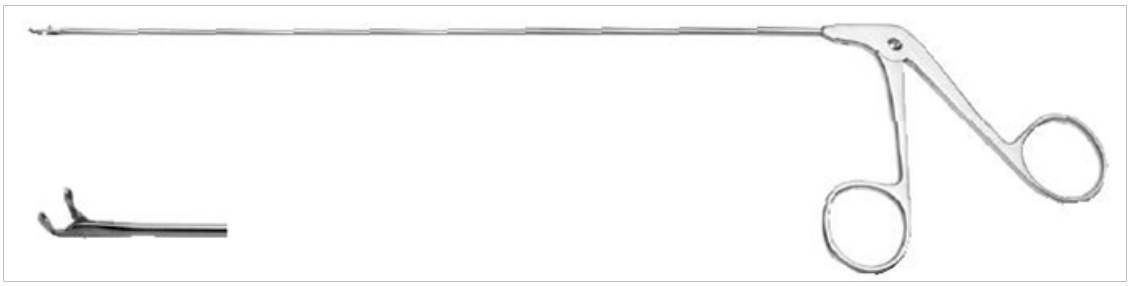

Figure 1.2 Image of Phonomicrosurgery Forceps. The tip of the instrument is shown in the lower left hand corner of the image.

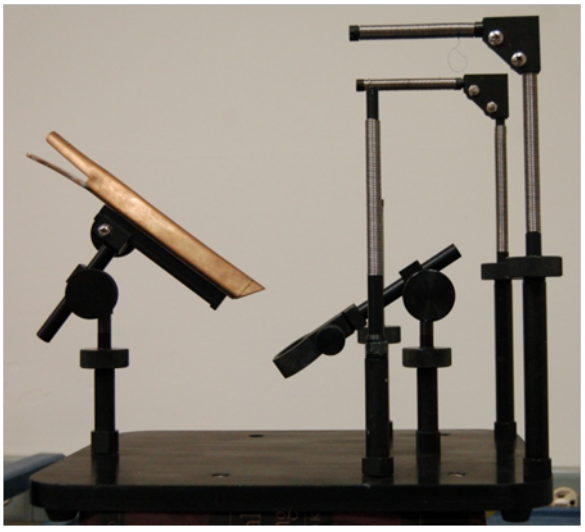

Figure 1.3 Laryngeal Dissection Station (LDS). A bronze laryngoscope is attached to the left of the base.

wired sensors connected to the tip of each instrument and a magnetic transmitter. The sensors received a signal from the transmitter that was used to measure their positions. Unfortunately, the position data reported by the tracking system was inconsistent with the actual instrument movements. Upon investigation it was found that the stainless steel surgical instruments were influencing the signals received by the sensors. This problem was the primary motivation for this thesis project.

1.2 Project Overview

A system named V-PITS (Video Based Phonomicrosurgery Instrument Tracking System) was developed to estimate the 3D trajectory of phonomicrosurgery instruments performing a simulated exercise. It was developed as an alternative to the magnetic position tracker described in the previous section. V-PITS consists of a data acquisition and algorithm component. The data acquisition component consists of two synchronized cameras. The algorithm component consists of two algorithms that are used to estimate the 3D trajectory of the phonomicrosurgery instruments from the video data. The two algorithms are a tracking algorithm and a trajectory estimation algorithm. The tracking algorithm tracks a set of features in each video. These features are related to the position and orientation of the instrument in the video. The trajectory estimation component combines the features from both videos to estimate the instrument's 3D trajectory.

Video was captured using the stereo camera rig seen in Figure 1.4. It consists of two Basler acA640-100gc machine vision cameras. An external trigger source is used for synchronization. The rig was integrated into a station designed to simulate phonomicrosurgery. Figure 1.5 is an image of the station. The LDS in Figure 1.3 is integrated into this setup. During an exercise a subject inserts instruments through the laryngoscope and manipulates a paper target. The camera rig is positioned within this station to capture the tips of the instruments as they manipulate the target. Figure 1.6 shows an example frame from both cameras within the rig. The two instruments and paper target are identified in the figure. A striped marker attached to each instrument is also identified. The tracking algorithm finds the position of this marker in

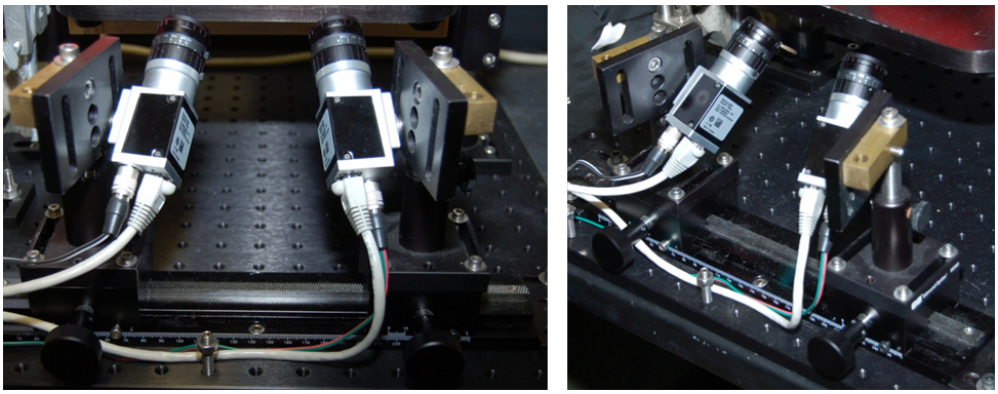

Figure 1.4 Stereo Camera Rig. Both cameras are mounted to mechanical devices that allows their position and orientation to be adjusted.

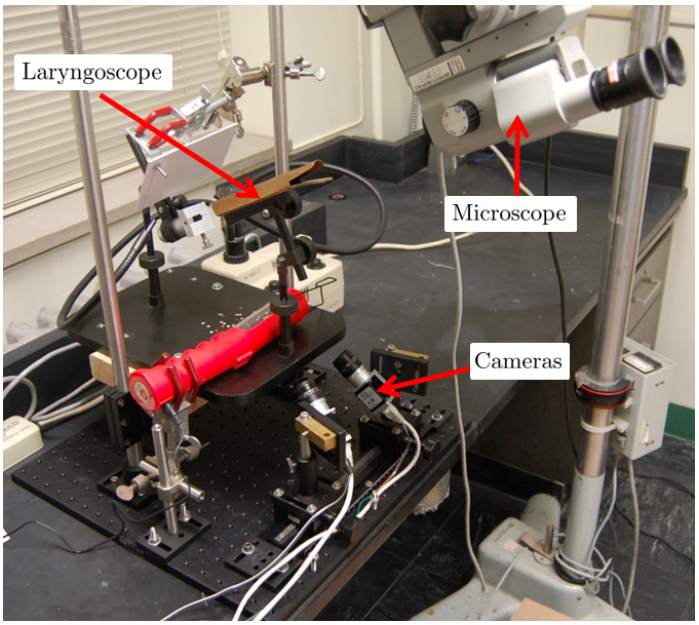

Figure 1.5 Phonomicrosurgery Station. A subject views a paper target using the surgical microscope. Surgical instruments are inserted into the laryngoscope to manipute the target. The cameras record video of the tips of the instruments manipulating the target.

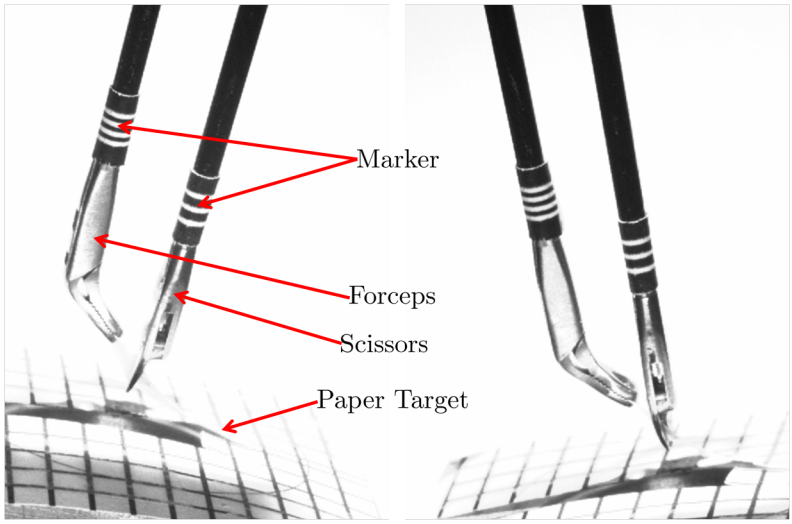

Figure 1.6 Left and Right Camera Example Frame. The instruments, attached markers, and paper target are identified in the left frame.

every frame. The rod of the instrument above the target is painted black. This provides high contrast at the boundaries of the instrument which are also tracked.

Figure 1.7 shows a sequence of frames in which a subject is attempting to grasp the paper target. In the first frame the forceps are being positioned near the paper target. The second frame shows the forceps grasping the task. In the third frame the target has been released by the forceps. The final frame shows the forceps re-grasping the target. A set of red overlays are seen on the forceps in every frame. The overlays represent features found by the tracking algorithm. Figure 1.8 shows plots of the forceps' 3D position over the course of the frame sequence. This data was generated using the trajectory estimation algorithm. The point at which the forceps grasps the target, releases the target, and re-grasps the target are identified in the plots.

1.3 Thesis Organization

The remainder of this document is organized as follows. Chapter 2 describes a methodology for estimating the three dimensional position of a point along an instrument's midline from images taken at two different viewpoints. Chapter 3 describes the instrument tracking

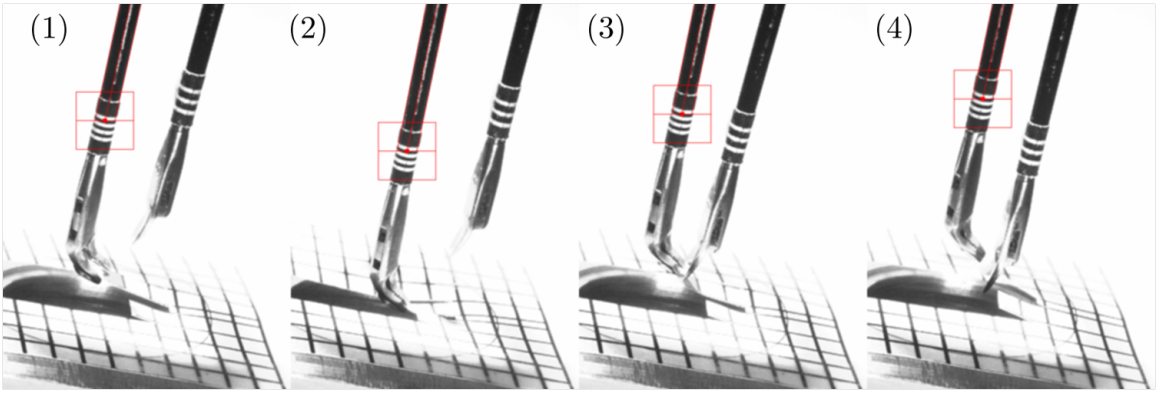

Figure 1.7 Grasping Frame Sequence. Image (1) shows the forceps being positioned prior to grasping the paper target. Image (2) shows the target being grasped. In Image (3) the target has been temporarily released. Image (4) shows the forceps re-grasping the target with the scissors in position to perform a cut. Red overlays in each image represent features found by the tracking algorithm.

algorithm. After capturing videos of an exercise, this algorithm is run on each video for each instrument. It detects a set of features in each video frame. These features are related to the position estimation methodology described in Chapter 2. The detected features in both videos are used to estimate the 3D trajectory of a surgical instrument. Chapter 4 describes the algorithm developed to do this. Chapter 5 describes experiments performed to characterize system noise and accuracy. Chapter 6 concludes this document and describes possible optimizations and future work.

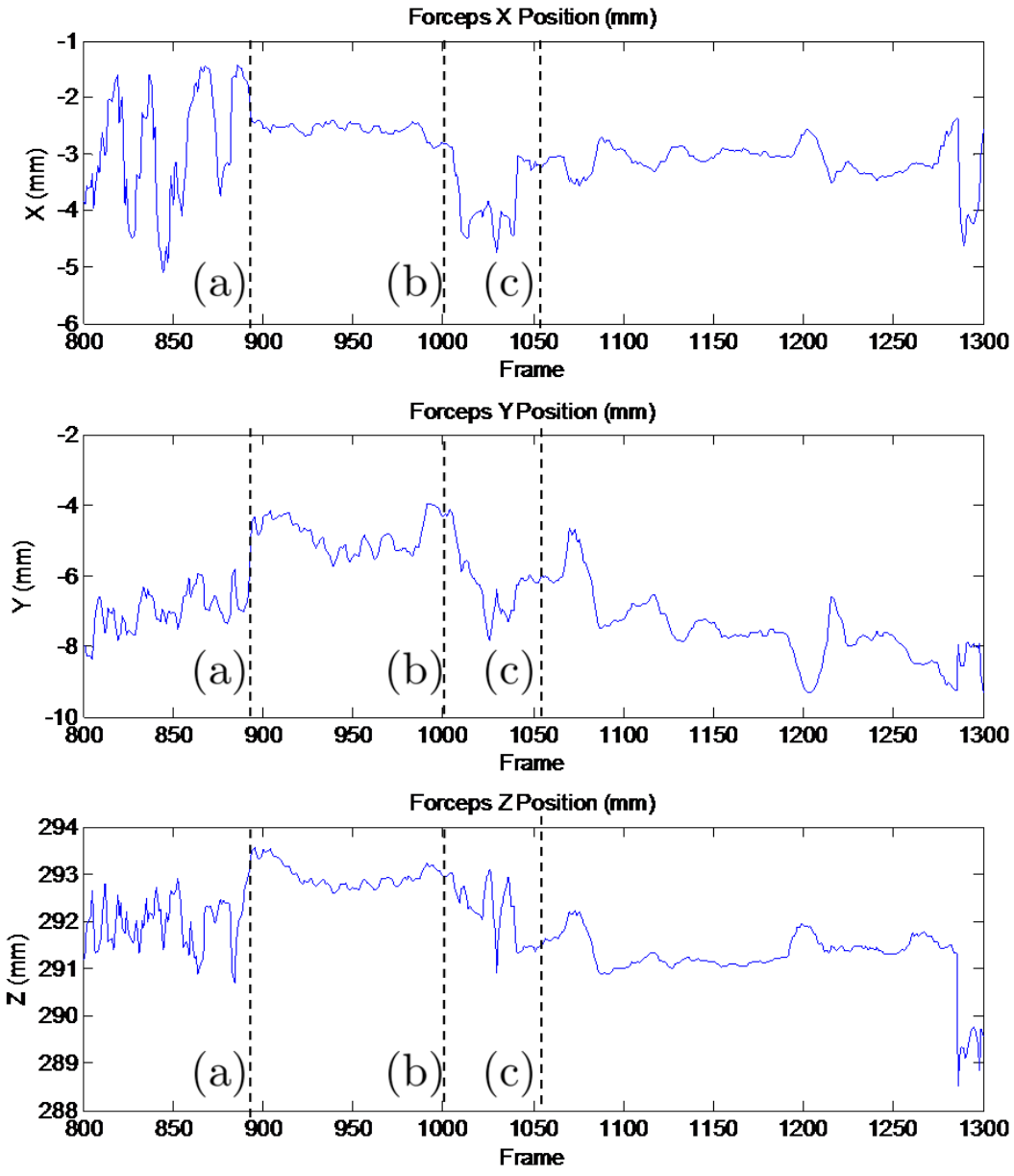

Figure 1.8 Frame Sequence Forceps Position Data. Initial grasping, release, and re-grasping events are identified in the plots by (a), (b), and (c) respectively.

Chapter 2

Surgical Instrument Position Estimation

The discrete trajectory of an object is given by its position at discrete points in time. Given a method for estimating position at a single point in time, trajectory can be found by repeatedly applying the method. This chapter describes a methodology for estimating the 3D position of a phonomicrosurgery instrument at a single point in time. It provides the basis for the tracking and trajectory estimation algorithms described in the following chapters.

Given two cameras viewing the same object at different viewpoints it is possible to estimate the position of a point on the object in 3D space. For example, assume it is desired to estimate the 3D position of a single interest point on an object given digital images captured by each camera. Assuming calibrated cameras, this consists of identifying an image point correspondence pair and triangulating position. A set of intrinsic or internal parameters of each camera and extrinsic parameters that define the spatial relationship between the two cameras define the calibration. An image point correspondence pair refers to one point in each image that corresponds to the 3D object interest point whose position is being estimated. The calibration parameters define a triangulation function that maps an image point correspondence pair to a 3D position. In Section 2.1 the pinhole camera model is described. The parameters within this model are the intrinsic parameters previously mentioned. Section 2.2 describes the geometric relationship of two cameras under the pinhole model. Position triangulation is introduced as a geometric concept and formulated algebraically using camera calibration parameters.

An image point correspondence pair is needed to triangulate position. A major challenge of this project was identifying this point pair for the surgical instrument. This is difficult due to the instrument's metallic texture and non-planar shape. In order to overcome this, the geometric

relationship between two views known as epipolar geometry is utilized as described in Section 2.2. Specifically, epipolar geometry along with knowledge of the phonomicrosurgery instrument's geometry is used to determine a point correspondence pair. Section 2.3.3 describes this concept and provides simulation results that validate components of it. Given this point correspondence pair the 3D position of a point on the surgical instrument can be estimated.

2.1 Camera Modeling and Single View Geometry

2.1.1 Action of a Camera

In this document a camera refers to a device that senses electromagnetic radiation in the visible light spectrum. Using a digital sensor it stores the light it senses in a digital image. Geometrically this image is a 2D representation of the light radiated from or reflected off of objects in the 3D world. Therefore the action of a camera can be modeled as a transformation from a point (the radiation or reflectance point) in the 3D world to a point on a 2D plane or image. A mathematical representation of this transformation is useful when information about the 3D world must be estimated from 2D images.

2.1.2 Pinhole Camera Model

The pinhole camera model [6] was used to represent the action of a camera in this project. Figure 2.1 is a geometric representation of this model. A Euclidean coordinate system with axes $X_{Frame}, Y_{Frame}, Z_{Frame}$ is defined with the camera center or center of projection, $\mathbf{C}$, as the origin and a plane called the image plane or focal plane at $Z_{Frame} = f$. This coordinate system is known as the camera frame. A coordinate system on the focal plane is defined with an origin at the principal point, $\mathbf{p}$. In Figure 2.1 $\mathbf{m}$ is the image of 3D world point $\mathbf{M}$ on the focal plane. The intersection between the focal plane and a line drawn from $\mathbf{M}$ to $\mathbf{C}$ defines the location of this point and is a geometric representation of the imaging process. Mathematically this is represented as

$$(X, Y, Z) \rightarrow (f\frac{X}{Z}, f\frac{Y}{Z}). \tag{2.1}$$

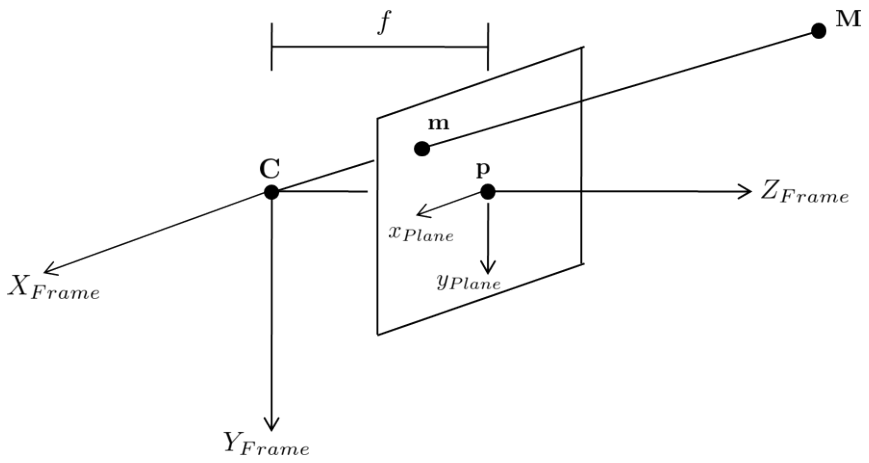

Figure 2.1 Camera Frame for Pinhole Model Geometry.

By utilizing a homogeneous representation of a 3D and 2D point, matrix multiplication can be used to represent imaging as

$$\begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \tag{2.2}$$

Whereas Equation (2.1) is a non-linear mapping between Euclidean spaces, $\mathbb{R}^3 \rightarrow \mathbb{R}^2$, equation (2.2) is a linear mapping between projective spaces, $\mathbb{P}^3 \rightarrow \mathbb{P}^2$. A point in N-dimensional Euclidean space corresponding to a point in N-dimensional projective space is found by dividing the first N elements of the homogeneous vector by its (N+1)th element. An example of this is

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} \rightarrow \begin{bmatrix} \frac{a}{c} \\ \frac{b}{c} \end{bmatrix} \tag{2.3}$$

for a point in 2D Projective and Euclidean space. A modification must be made to the pinhole model described because the camera generates a digital image. The parameter f represents a distance in the camera frame, therefore the image plane coordinates have metric units. These units are representative of position on the imaging sensor. Yet an image is quantized into units

of pixels. Therefore a multiplicative factor is needed to convert spatial units. The origin of a digital image is commonly assigned to the pixel in the upper-left corner. This is accounted for by including a translation into the model. Additionally, a skew factor is included into the model to account for non-square pixels. The Euclidean non-linear mapping for this complete model is given by

$$(X, Y, Z) \rightarrow (\alpha_x \frac{X}{Z} + s\frac{Y}{Z} + u_0, \alpha_y \frac{Y}{Z} + v_0). \tag{2.4}$$

The parameters α_x and α_y incorporate f and a conversion factor to get pixel units. The digital image origin is accounted for with parameters u_0 and v_0 and parameter s accounts for skew. This can once again be represented as a linear projective mapping as

$$\begin{bmatrix} \alpha_x & s & u_0 & 0 \\ 0 & \alpha_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathbf{P}\tilde{\mathbf{X}}. \tag{2.5}$$

The (3x4) matrix giving this mapping is known as the projective camera matrix $\mathbf{P}$. In the next section this matrix will be decomposed in order to describe intrinsic camera calibration parameters.

2.1.3 Intrinsic Parameters

The camera matrix $\mathbf{P}$ can be decomposed into the multiplication of a (3x3) matrix with a (3x4) matrix. This decomposition is given by

$$\mathbf{P} = \begin{bmatrix} \alpha_x & s & u_0 & 0 \\ 0 & \alpha_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} \alpha_x & s & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \mathbf{K} \left[I_{3x3} | \mathbf{0} \right] \tag{2.6}$$

where the (3x3) matrix $\mathbf{K}$ is known as the intrinsic camera calibration matrix. The (3x4) matrix acts as a projection from the 3D camera frame to a 2D plane. It provides the mapping $\mathbb{P}^3 \rightarrow \mathbb{P}^2$. $\mathbf{K}$ acts to transform the coordinate system of the plane such that it is consistent with the digital image captured by the camera. It contains the internal parameters of the camera

$\{\alpha_x, \alpha_y, s, u_0, v_0\}$ while no parameters are within the (3x4) matrix. These internal parameters are known as the intrinsics of a camera (under the pinhole model), and intrinsic calibration is a process that estimates these parameters.

2.1.4 Extrinsic Parameters

The previous subsections defined a transformation from a 3D camera frame to an image. Coordinates of points in 3D space were assumed in a frame with the camera center at the origin. This is important because it allows image formation of a point to be represented as the intersection of a line between the point and the camera center with a plane normal to the Z-axis. If the position of a point in 3D space is given with respect to a different coordinate system, a transformation must be applied to find its position in the camera frame. When dealing with multiple cameras (and correspondingly multiple camera frames) this concept becomes important. Conveniently, this coordinate transformation can be included in the camera matrix.

Assume the 3D position of a point is given in a coordinate frame known as the world frame. Additionally, the rigid body transformation between the world frame and the camera frame is known. The rigid body transformation [8] consists of two components: translation and rotation. Translation is defined by the vector between the world frame origin and the camera center (origin of the camera frame). Rotation is defined by the relative orientation of the camera frame with respect to the world frame. This is represented as

$$\mathbf{X}_{Camera} = \mathbf{R}\mathbf{X}_{World} + \mathbf{t} \tag{2.7}$$

where $\mathbf{X}_{World}$ is a point in the Euclidean world frame, $\mathbf{R}$ is a (3x3) orthogonal matrix defining the rotation, $\mathbf{t}$ is a (3x1) vector, and $\mathbf{X}_{Camera}$ is the same point in the camera frame. Figure 2.2 illustrates this transformation. Note that given knowledge of the rotation matrix, $\mathbf{R}$, and knowledge of the camera center in the world frame, $\mathbf{C}$, the translation is found as $\mathbf{t} = -\mathbf{R}\mathbf{C}$. The two components of the transformation can be included directly in the camera matrix as

$$\mathbf{P} = \mathbf{K}\left[\mathbf{R}|\mathbf{t}\right]. \tag{2.8}$$

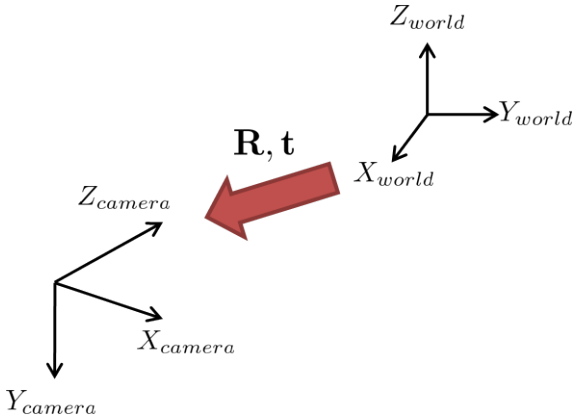

Figure 2.2 Rigid Body Transformation Between World Frame and Camera Frame.

Using this form of the camera matrix models imaging of 3D points in the world frame by a camera whose position and orientation in the world frame is defined by $\mathbf{R}$,$\mathbf{t}$. In this case the world point is homogeneous as $\mathbf{P}$ provides a mapping between projective spaces.

The parameters $\mathbf{R}$ and $\mathbf{t}$ are known as the extrinsic parameters of the camera. Position and orientation of the camera with respect to some world frame is defined by them. Extrinsic calibration is a process of estimating these parameters. The importance of these parameters will be seen in Section 2.2 in which two cameras in a world frame are analyzed.

2.1.5 Ray Back-Projection

Figure 2.1 shows that the pinhole model geometrically represents image formation in 3D space as the intersection of a plane with a line connecting a point with the camera center. Given a point in an image captured by a camera with known intrinsic and extrinsic parameters, a ray along this line can be calculated. The calculation of a ray using an image point location is known as back-projection. A point on this ray, $\mathbf{l}(\lambda)$, in the world frame is given by

$$\mathbf{l}(\lambda) = \mathbf{P}^{\dagger}\tilde{\mathbf{u}}_{image} + \lambda\tilde{\mathbf{C}} \tag{2.9}$$

where λ is a free scalar parameter, $\mathbf{P}^{\dagger}$ is the pseudo-inverse of the camera matrix, $\tilde{\mathbf{u}}_{image}$ is the image point being back-projected in homogeneous coordinates, and $\tilde{\mathbf{C}}$ is the homogeneous coordinate position of the camera center in the world frame. The back-projected ray provides a

constraint in 3D space for the position of a point corresponding to an image point. In order to extract its 3D position, additional constraints are needed. In the next section two cameras are introduced. By incorporating constraints from an additional camera it is possible to estimate 3D position.

2.2 Stereo Camera Geometry and Point Triangulation

V-PITS uses two cameras to simultaneously capture video of phonomicrosurgery instruments. Underlying the choice of this stereo configuration is 3D triangulation. Given two calibrated views of a point, the 3D position of the point with respect to the views can be estimated. This section discusses the geometry of a stereo camera configuration under the pinhole model described in Section 2.1.

2.2.1 Correspondence and Triangulation

Figure 2.3 is a scene with two cameras viewing the same point, $\mathbf{X}$. Both cameras are modeled using the pinhole model and $\mathbf{X}$ is imaged as $\mathbf{u}_1$ and $\mathbf{u}_2$ by each respective camera. Subsection 2.1.4 described how points in a world reference frame could be transformed to a camera frame given knowledge of a rigid body transformation between the two frames. For a stereo configuration as seen in Figure 2.3 this is very important. Here the world frame is assumed aligned with one of the camera frames. Therefore the external parameters (rigid body transformation) can be used to transform a point's location from one camera frame to another. In a stereo camera configuration, the rigid body transformation parameters $\{\mathbf{R}, \mathbf{t}\}$ for this transformation between camera frames are known as the extrinsic calibration parameters. These parameters describe the spatial relationship between the two cameras. In the next paragraphs triangulation is discussed. The extrinsic calibration parameters are utilized to incorporate information from both camera frames to make an estimate of 3D position.

Figure 2.3 shows a point $\mathbf{X}$ in the 3D world being viewed by two cameras. Here the imaging process is illustrated by drawing a line from $\mathbf{X}$ to each camera center. The 2D point pair $\mathbf{u}_1$ and $\mathbf{u}_2$ are known as an image correspondence pair. They are in correspondence because

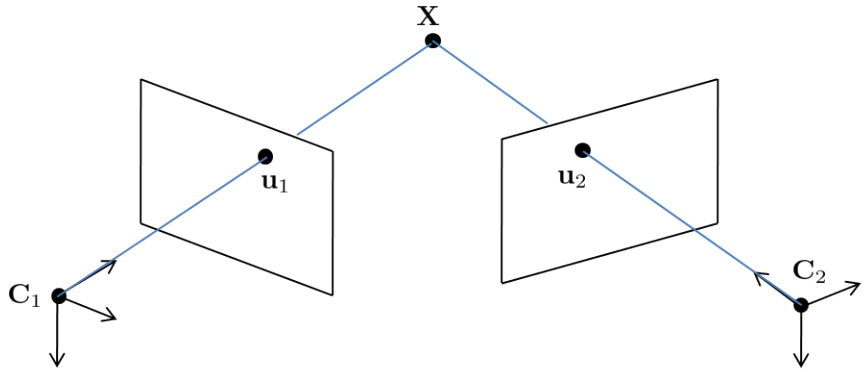

Figure 2.3 Two Cameras Imaging the Same Point $\mathbf{X}$ in 3D space.

they are the images of the same point in 3D space. Given $\mathbf{u}_1$ and $\mathbf{u}_2$, their corresponding 3D point $\mathbf{X}$ can be found by locating the intersection of the back-projected (Subsection 2.1.5) ray from each point. This requires knowledge of intrinsic and extrinsic calibration parameters.

Estimation of 3D position from a correspondence pair is known as triangulation. One issue with this geometric method of triangulation is that noise or error in correspondence point locations can lead to rays that do not intersect. To overcome this issue when performing triangulation the midpoint algorithm can be used. Figure 2.4 is a block diagram describing the algorithm and Figure 2.5 is an illustration of it. Instead of locating the point of intersection, a line is drawn between the points on each back-projected ray that are nearest to one another. The midpoint of this line is taken as the 3D point corresponding to the correspondence pair. Therefore regardless of noise or error in the correspondence pair, a location in 3D space is found.

The midpoint algorithm provides for 3D point triangulation given a correspondence pair. Yet prior to triangulation, a correspondence pair must be extracted from the camera images. This correspondence problem can be formalized as: given a point $\mathbf{u}_1$ from camera 1 identify a point in camera 2 that is the image of the same 3D world point as $\mathbf{u}_1$. This is an unconstrained problem, because it allows any point in camera 2 to be matched with $\mathbf{u}_1$. The next section describes a geometric constraint on this search for a correspondence point known as the epipolar constraint.

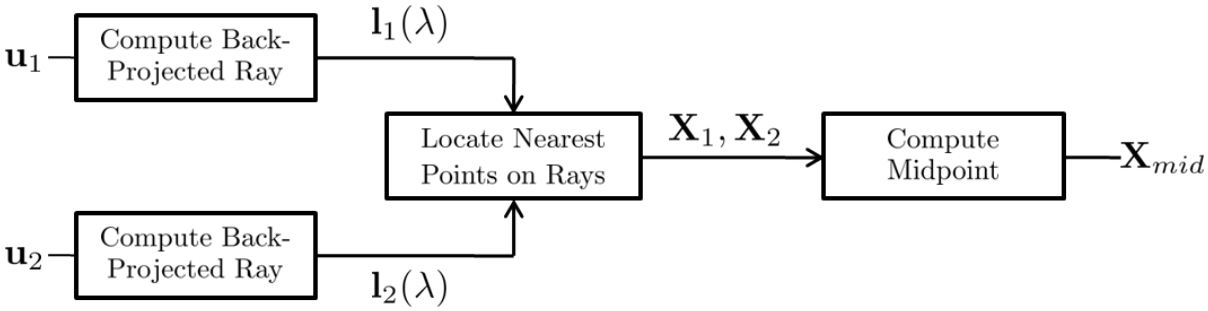

Figure 2.4 Midpoint Triangulation Algorithm Block Diagram.

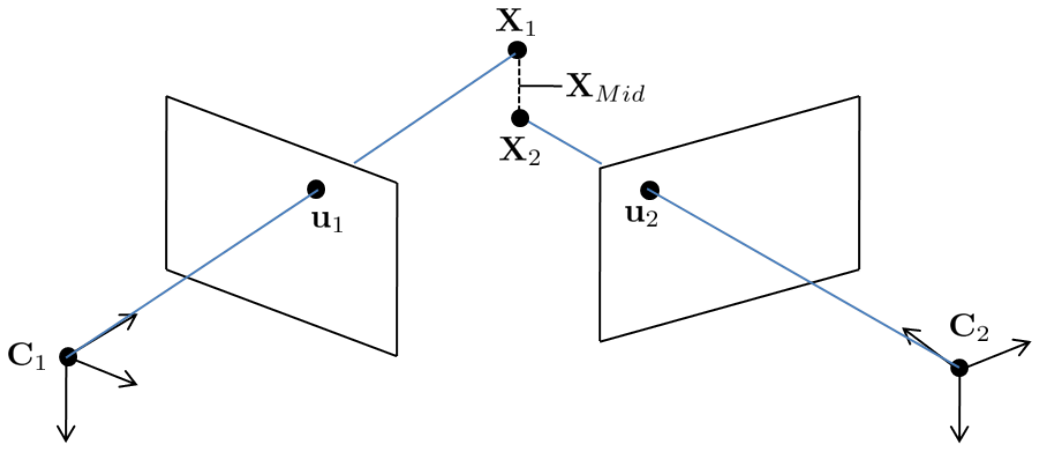

Figure 2.5 Midpoint Triangulation Algorithm Illustration.

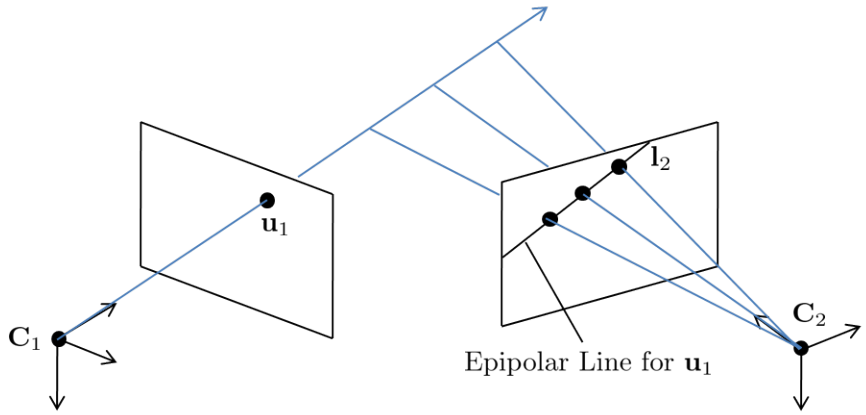

Figure 2.6 The Epipolar Constraint.

2.2.2 The Epipolar Constraint

An illustration of the epipolar constraint is given in Figure 2.6. Here the back-projected ray through camera one's image point $\mathbf{u}_1$ is shown. Points on this ray are possible 3D world points that $\mathbf{u}_1$ is an image of. Therefore correspondence points in camera 2 must be located on the image of this back-projected ray in camera 2, which is a line. Thus, the location of a point in camera 2 corresponding to a point in camera 1 is constrained to a line. With this constraint the search for a correspondence point has been reduced from two dimensions to one dimension. Given a point in camera 1, a search for a correspondence point in camera 2 occurs over a line known as an epipolar line. The epipolar constraint establishes a relationship between a point in one camera and a line in another.

Algebraically this is represented by the Longuet-Higgins equation [2] given by

$$\tilde{\mathbf{u}}_2^T \mathbf{F} \tilde{\mathbf{u}}_1 = 0. \tag{2.10}$$

When $(\tilde{\mathbf{u}}_1, \tilde{\mathbf{u}}_2)$ is a correspondence pair in homogeneous coordinates that satisfy the epipolar constraint, this equation is satisfied. $\mathbf{F}$ is the fundamental matrix, which is a (3x3) rank 2 matrix. Geometrically it maps a point in one camera image to its epipolar line in the other. This mapping is given by

$$\tilde{\mathrm{l}}_2 = \mathbf{F}\tilde{\mathbf{u}}_1, \tilde{\mathrm{l}}_1 = \mathbf{F}^T\tilde{\mathbf{u}}_2 \tag{2.11}$$

where $\tilde{l_2}$ is the epipolar line corresponding to point $\tilde{u}_1$ in camera 1, and $\tilde{l_1}$ is the epipolar line in camera 1 corresponding to point $\tilde{u}_2$ in camera 2. For an explanation of line representation in 2D homogeneous coordinates see Subsection 3.1.4 in Chapter 3.

This subsection introduced the epipolar constraint. Previously it was seen that a point correspondence between two cameras can be used to estimate the 3D position of a point. Given a point in one camera, the epipolar constraint reduces the search for a correspondence point in the other camera from the entire image to a line. Yet the question still remains, how to select the appropriate point on this epipolar line? Ideally, the correct correspondence pair consists of two points that are the image of the same 3D world point. The next section discusses this topic with respect to a phonomicrosurgery instrument. A method for finding a point correspondence pair is introduced. This method utilizes geometric object features along with the epipolar constraint to generate a correspondence pair belonging to a point on the surgical instrument.

2.3 Point Correspondence and Cylindrical Instruments

2.3.1 Issues with Similarity Based Point Correspondence Techniques

The previous section introduced triangulation for estimating 3D position given a calibrated camera configuration. In order to triangulate position, an image point correspondence pair between cameras is needed. The points in this correspondence pair are taken to be images of the same point in the 3D world. If one point in the correspondence pair is given, the epipolar constraint can be used to reduce the search for the other point to a line. A single point on this line must be selected based on the higher level concept that both points are images of the same 3D world point. One common technique is to find the image point on the epipolar line that is maximally similar to the given image point.

Two possible measures of similarity are the normalized cross-correlation coefficient and sum of squared differences. Both rely on comparisons of image data in a window around the points being compared. These methods rely on the following two properties: 1) the windows must have significant enough texture to be matched and 2) the same image windows must be visible to both cameras. Property 1 is needed to make the matching process robust

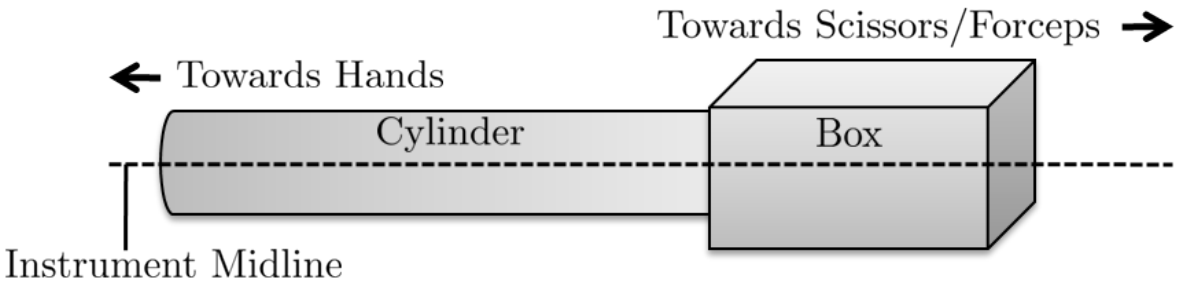

Figure 2.7 Simple Geometric Phonomicrosurgery Instrument Model.

to image noise and spatial lighting variations. The metallic texture of the surgical instruments presented two issues with respect to property 1. Instrument surfaces were not highly textured and exhibited specular reflections which were dependent on the position and orientation of the instrument. This was accounted for by attaching a marker to the instruments with a distinct pattern/texture and Lambertian reflectance properties.

Property 2 is a constraint on the geometry between the object being viewed and the cameras. A basic geometric instrument model is seen in Figure 2.7. Two non-planar shapes, a box and a cylinder, make up this model. Surfaces of both shapes can go in and out of view due to instrument-axis rotations, which can lead to property 2 being violated. Additionally, rotation about the instrument midline causes the image of surfaces on the instrument to translate. Therefore extraction of motion information is difficult because there is ambiguity in knowing whether the instrument was translated or rotated about its midline.

The algorithms in V-PITS use an alternative approach based on assumptions about the instrument's geometry (specifically the cylindrical portion). Geometric features are extracted from the two images of the instrument in order to estimate the imaged location of each instrument's midline. Based on this midline in both images and the epipolar constraint, a point correspondence pair is found. The following two subsections describe this concept in more detail. Subsection 2.3.2 discusses the extraction of the imaged midline location in an image of a circular and cylindrical object. Results of a simulation are presented as a verification of the concept. Subsection 2.3.3 describes how the imaged midlines can be used in conjunction with the epipolar constraint to form a point correspondence pair.

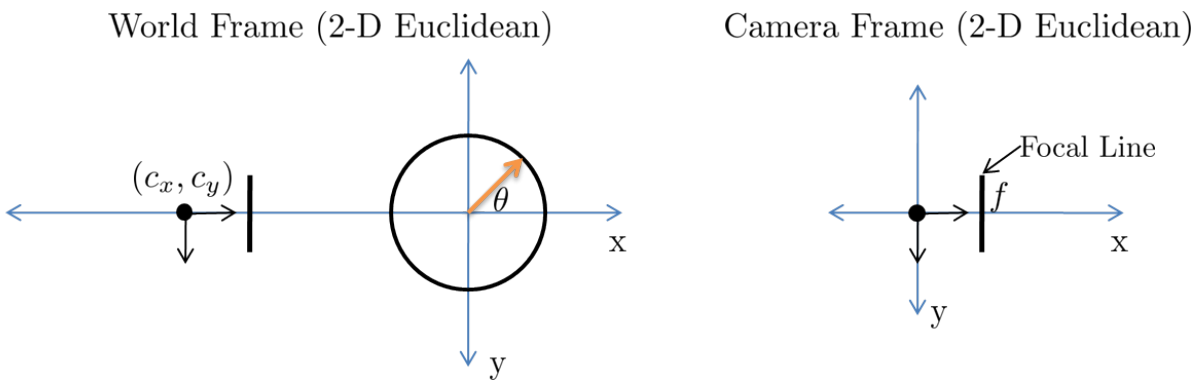

Figure 2.8 Camera Viewing Circle in Two Dimensional World Frame.

2.3.2 Imaged Midline Estimation

Imaged midline estimation is based on the following idea: the location at which an object's midline would be imaged can be estimated based on the location of the object's visible boundaries. Because the object's midline is a geometric concept, it cannot be seen directly. Assuming contrast between the object and the background, the imaged locations of the object's boundaries can be found using edge detection techniques. Then, based on prior knowledge of the object's geometry, the midline location can be estimated. The following paragraph presents this concept in two-dimensions for a circular object.

Figure 2.8 is an illustration of a 2D world frame with a camera is viewing a circle. Here the camera projects points in the 2D world onto a line based on the pinhole camera model. The goal is to estimate the position of the imaged center of the circle from the location of the visible boundaries. A simple estimate for the center location is given by

$$\hat{u}_{center} = \frac{u_{leftBound} + u_{rightBound}}{2} \tag{2.12}$$

where $u_{leftBound}$ and $u_{rightBound}$ are the locations of the visible boundaries and $\hat{u}_{center}$ is the estimated location of the image of the circle's center.

To evaluate the accuracy of this estimation method a simulation was performed. Imaging was simulated using a pinhole model camera in two dimensions with a focal length of $f = 25$ mm. This was selected because the lenses in the stereo camera rig had a focal length of 25 mm. Because only a focal length parameter was utilized to parameterize the intrinsics,

Parameter	Value(s)
f	25 mm
D	1.93 mm
c_x	-300:1:-200 mm
c_y	-50:1:50 mm

Table 2.1 Circle Center Estimation Simulation Parameters

the imaging process (2D projection onto 1D) resulted in a physical position on a focal line analogous to the focal plane in three dimensions. The world coordinate frame in Figure 2.8 was utilized. A circle of diameter $D = 1.93$ mm was located at the origin of this coordinate system. This diameter was chosen based on the diameter of the cylindrical portion of the phonomicrosurgery instrument. The location of the camera center was varied based on the expected camera location with respect to the surgical instruments. The x location of the camera center, c_x, was varied from -300 mm to -200 mm at 1 mm increments. The y location, c_y, was varied from -50 mm to 50 mm at 1 mm increments. Table 2.1 shows the parameters used in the simulation. For every combination of parameters, imaging of the circle was simulated. The circle was discretized into 2000 radially spaced points. A position on the focal line was calculated for each point based on the pinhole model. Left and right boundary points were assigned as extrema on the focal line and an estimate of the center location was found using Equation (2.12). Finally, the error in this estimate was calculated as the difference between the estimate and the imaged location of the actual circle center (located at the origin of the world coordinate system).

Figure 2.9 is a graph of the estimation error as a function of camera center position. The x-axis is the value of parameter c_x and the y-axis is the value of c_y. Graph color indicates estimation error in mm. A maximum error of 5.82 nm was found. In this simulation the focal line was analogous to the focal plane in three dimensions. When only a focal length intrinsic parameter is used, position and distances on the focal plane are representative of those on the imaging sensor. Therefore the error was compared to the dimensions of a single pixel

on an imaging sensor. Specifically, in this project an imaging sensor with pixels of dimension 5.6 μm x 5.6 μm was used. Therefore the maximum center estimation error was approximately 0.1 % of a pixel side-length. This provided strong evidence that this type of estimation method could be utilized as the error was well below the image's resolution.

A 3D simulation using a cylindrical object was also performed to verify the estimation method. Figure 2.10 is a graphic of a 3D world frame with a camera viewing a cylinder. We now want to estimate the position of the imaged midline of this cylinder using its visible boundaries. The previous method used two boundary points to estimate a center point. This method was extended by replacing points with lines. A diagram of the method used is seen in Figure 2.11. Once again a simple average is used, but instead of points it is an average of lines. First, parameterized boundary lines $(\rho_{left}, \theta_{left})$ and $(\rho_{right}, \theta_{right})$ are fit to the imaged boundary points. A parameterized estimate of the midline $(\hat{\rho}_{mid}, \hat{\theta}_{mid})$ is found by averaging these two lines.

The accuracy of this estimation method was evaluated using a simulation. The previous simulation varied the camera center location with respect to a circle centered at the origin of the world frame. Figure 2.10 illustrates the simulation setup used. Here a cylinder is centered on the Z-axis of the world frame. A cylinder height h = 30 mm and diameter D = 1.93 mm was used. Imaging was simulated using a pinhole model camera in three dimensions with a focal length f = 25 mm. The camera center position in the x-y plane, c_x and c_y, tilt, t_θ, and roll, r_θ, of the camera frame were parameterized. Once again parameter values were chosen based on actual equipment that would be utilized and the expected geometry between the phonomicrosurgery instruments and cameras. Table 2.2 gives the parameter values used in the simulation. For every combination of parameters, imaging of the cylinder was simulated. The cylinder was discretized into a set of 100 evenly spaced discs along its height. Each disc was discretized into 2000 radially spaced points. A set of boundary points was associated with each disc by locating the horizontal extrema of the imaged disc. Imaging of each disc resulted in a set of boundary points. Boundary lines were fit to the boundary points and used to estimate the imaged midline. Next, points on the actual cylinder midline were imaged.

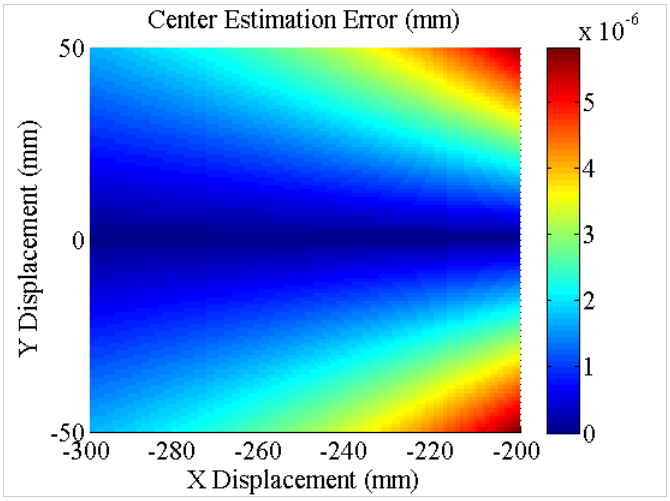

Figure 2.9 Simulated Center Estimation Error as a Function of Camera Center Location.

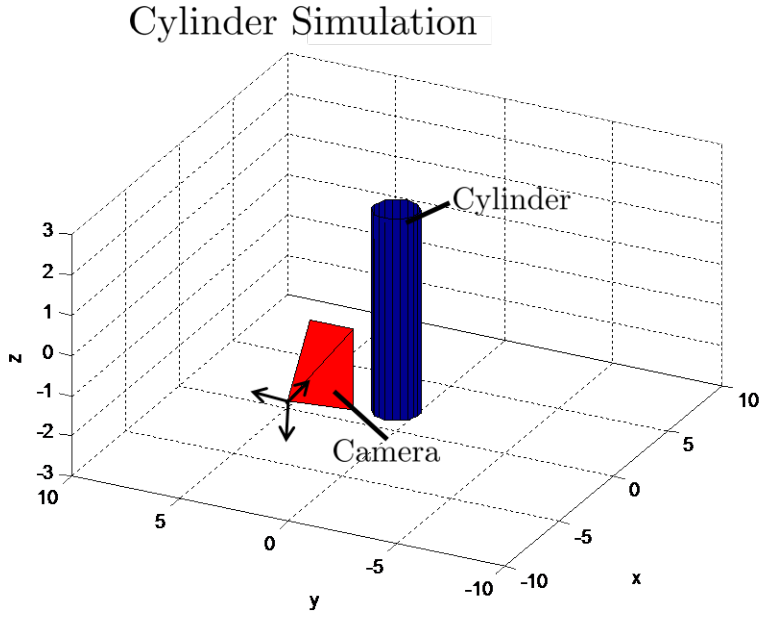

Figure 2.10 Camera Viewing Cylinder in Three-Dimensional World Frame.

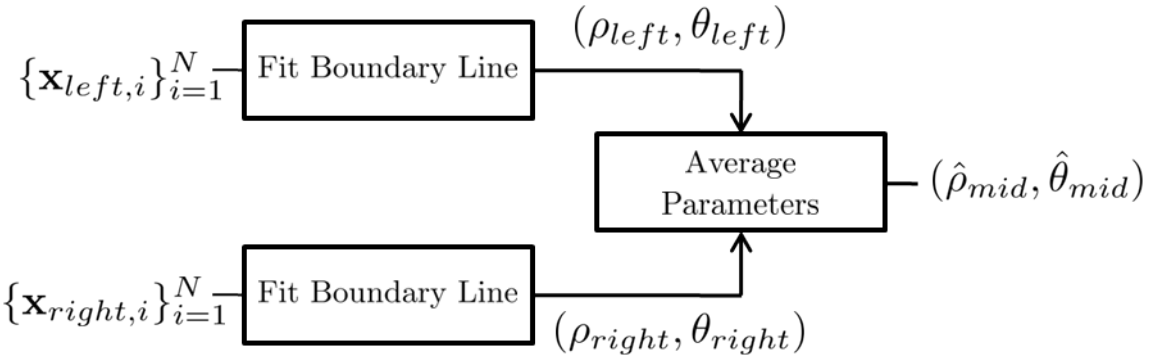

Figure 2.11 Midline Estimation Method Block Diagram

Deviation between these points and points on the estimated midline were calculated. Error in the midline estimate was taken as the maximum deviation.

Figure 2.12 shows midline estimation error as a function of camera center position for fixed roll and tilt values. Graph color indicates estimation error in mm. Similar spatial error behavior is seen in all graphs, showing that camera center position had the greatest influence on estimation error for this parameter set. The worst error occurred at the minimum in magnitude x-position and maximum in magnitude y-position in all three graphs. The maximum midline estimation error observed over the parameter set was approximately 150 nm. This is 2.7% of the side-length of a pixel in the imaging sensor used in this project. Therefore the geometric error introduced by this estimation method is much lower than an image's pixel resolution and validates the use of the method. This midline estimate is called the imaged midline.

A simplistic model of a phonomicrosurgery instrument is given in Figure 2.7 where the rod of the instrument is represented as a cylinder. The estimation methods introduced in the previous paragraphs can be used to estimate the rod's imaged midline in an image of the instrument. The primary goal of this chapter was to introduce a technique for estimating the 3D position of a point on a phonomicrosurgery instrument using a calibrated stereo camera rig. It was explained earlier that this required finding an imaged point correspondence pair between the two views. Given a point on the imaged midline in one view the next section shows that the other point in the correspondence pair is simply the intersection of the imaged midline and epipolar line in the other view.

Parameter	Value(s)
f	25 mm
D	1.93 mm
h	30 mm
c_x	-300:10:-200 mm
c_y	-50:10:50 mm
t_θ	$-\frac{\pi}{8} : \frac{\pi}{16} : \frac{\pi}{8}$
r_θ	$-\frac{\pi}{8} : \frac{\pi}{16} : \frac{\pi}{8}$

Table 2.2 Cylinder Midline Estimation Simulation Parameters.

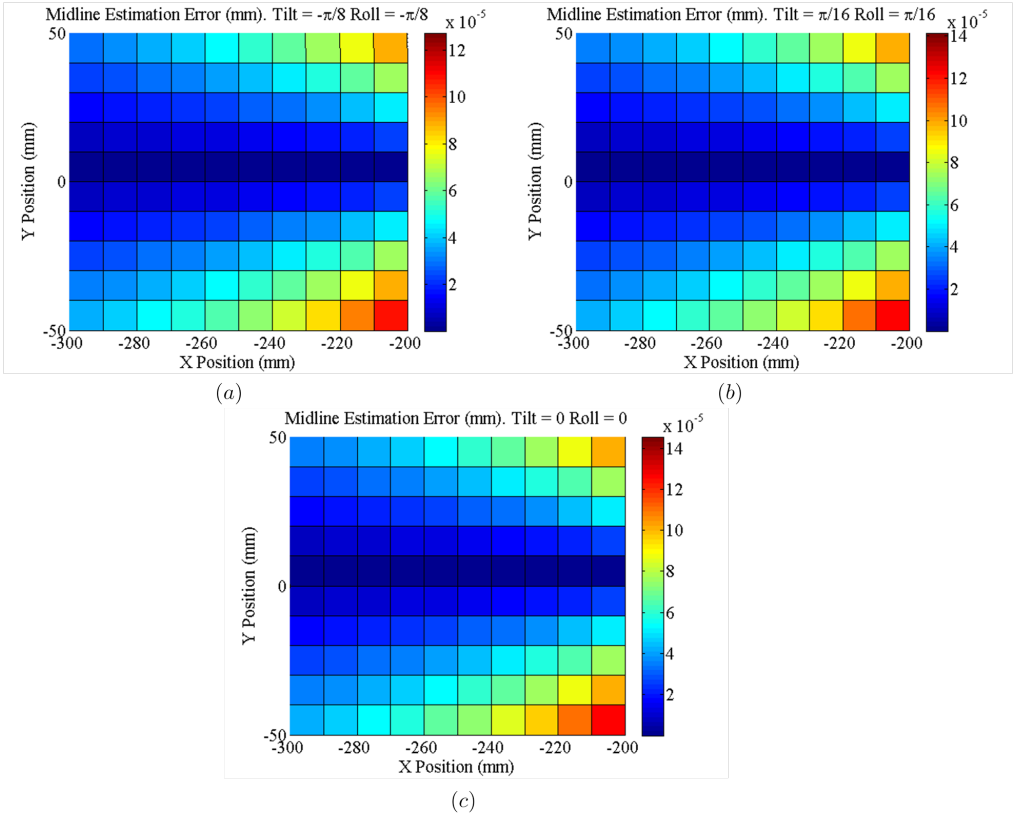

Figure 2.12 Midline Estimation Error as a Function of Camera Center Position. Results shown for tilt and roll angle fixed at (a) -$\frac{\pi}{8}$ (b) $\frac{\pi}{16}$ (c) 0.

2.3.3 Surgical Instrument Point Correspondence

Assume a calibrated stereo camera configuration is viewing a phonomicrosurgery instrument. Additionally assume image processing techniques have been utilized to estimate the imaged midline of the cylindrical rod in each view. The goal is to triangulate the position of a point on the imaged midline, which requires a point correspondence. Note that the imaged midline in each view is an estimate of the same line in 3D space, the true instrument midline. Therefore a point on one midline in one view corresponds to some point on the midline in the other view. Additionally, from the epipolar constraint, it is known that a point in one view corresponds to a point on the epipolar line in the other view. Therefore given a point on the imaged midline in one view, the intersection of the imaged midline and epipolar line define its correspondence point in the other view. Figure 2.13 illustrates this concept.

Algebraically this point correspondence pair can be found as

$$\tilde{\mathbf{u}}_{intersect} = \tilde{\mathbf{l}}_1 \times \tilde{\mathbf{l}}_2. \tag{2.13}$$

This equation defines the intersection point $\tilde{\mathbf{u}}_{intersect}$ of two lines $\tilde{\mathbf{l}}_1$ and $\tilde{\mathbf{l}}_2$ in 2D homogeneous coordinates. A homogeneous line representation can be defined given a (ρ, θ) line parameterization as

$$\tilde{\mathbf{l}} = \begin{bmatrix} cos(\theta) & sin(\theta) & -\rho \end{bmatrix}^T. \tag{2.14}$$

Let $\tilde{\mathbf{u}}_{source}$ be a point on the imaged midline in camera image 1, $\tilde{\mathbf{l}}_{mid}$ be the imaged midline in camera image 2, and $\mathbf{F}$ the fundamental matrix. The correspondence point in camera image 2, $\tilde{\mathbf{u}}_{correspond}$, is found using

$$\tilde{\mathbf{u}}_{correspond} = \mathbf{F}\tilde{\mathbf{u}}_{source} \times \tilde{\mathbf{l}}_{mid}. \tag{2.15}$$

Note that the points and lines in this equation are in homogeneous coordinates.

2.3.4 Requirements to Implement

This section has described a method for finding an image point correspondence pair for a 3D point on the midline of a phonomicrosurgery instrument. The 3D position of the midline

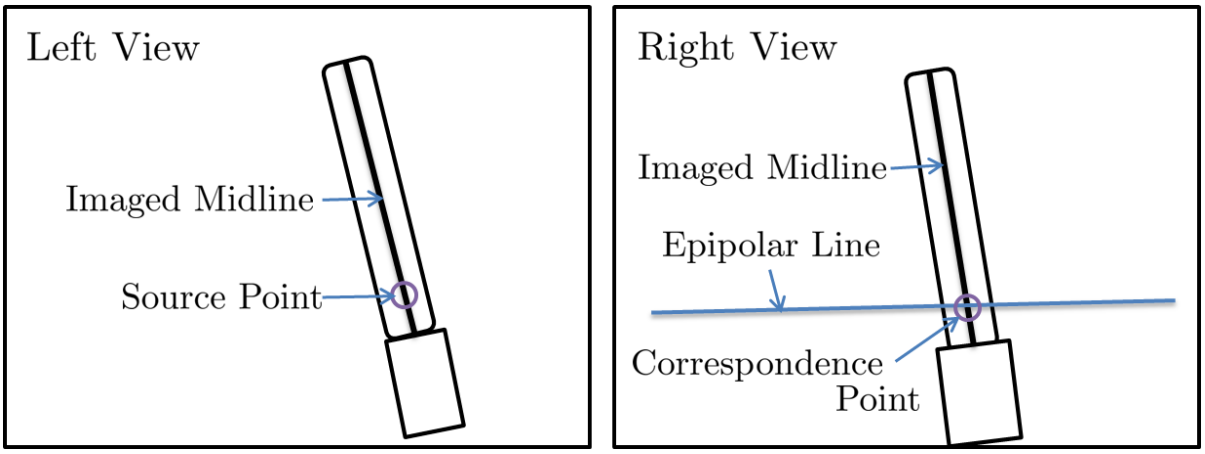

Figure 2.13 Correspondence Via Intersection of Epipolar Line and Imaged Midline. The source point lies on the imaged midline. It defines an epipolar line in the right view. A correspondence point in the right view is found by locating the intersection of the epipolar line with the imaged midline.

point is estimated from the correspondence using the midpoint triangulation algorithm. This has been a high-level treatment of the method and there are several details that need to be addressed in terms of implementation. Chapters 3 and 4 describe the algorithms based on this methodology used to estimate 3D instrument trajectory.

It has been assumed that the stereo camera configuration viewing the instrument is calibrated. A procedure must be performed to estimate each camera's intrinsic and the extrinsic parameters. This procedure is discussed in Subsection 4.1.2 of Chapter 4. Lines representative of the visible instrument boundaries are used to make an estimate of the imaged midline. Subsection 3.2.2 of Chapter 3 presents the algorithm used to extract these boundary lines from the video data. Finally, the point correspondence pair finding method assumed that a point on the imaged midline is given in one view. Yet it was not mentioned how this point is selected. Chapter 3 addresses this by incorporating imaged midline estimation into a tracking framework.

2.4 Chapter Conclusion

A method for estimating the 3D position of a point on a phonomicrosurgery instrument has been described. It is based on the pinhole camera model. The geometry between two pinhole modeled cameras provides for triangulation of 3D position based on back-projected rays. A point correspondence pair between camera images is needed to perform triangulation. Simulated data was presented to validate a method for estimating the imaged midline in an image of a phonomicrosurgery instrument. By incorporating the imaged midline with the epipolar constraint a point correspondence pair can be found based on the intersection of two lines. Finally, 3D position is estimated using the point correspondence pair and the midpoint triangulation algorithm.

The methodology in this chapter serves as the basis for the tracking and trajectory estimation algorithm used within V-PITS. The tracking algorithm detects the instrument's imaged midline and a point on this midline in every frame of a monocular video. The trajectory estimation algorithm applies this position estimation methodology at each video frame. The tracked feature data is used to form a point correspondence pair and 3D position is triangulated using the midpoint algorithm.

Chapter 3

Instrument Tracking Algorithm

V-PITS was developed to estimate the three-dimensional trajectory of up to two phonomicrosurgery instruments performing a simulated surgical exercise. During an exercise, a calibrated stereo camera rig captures a left and right pair of videos of the instruments. Afterwards, an instrument tracking algorithm is applied to both videos. This algorithm tracks a set of instrument features in every frame of the video. These features are used to estimate the three-dimensional trajectory of each instrument. This chapter details the algorithm used within V-PITS to track an instrument's features in a monocular video (monocular video means a video from a single camera in the stereo camera rig).

The first section introduces image processing techniques used in the algorithm. The second section describes the tracking algorithm. Due to blur and occlusion the algorithm can fail. To overcome this, the tracking algorithm is used within a user-guided scheme. This scheme is based on a measure of algorithm confidence and described in the final section.

3.1 Tracking Algorithm Image Processing Background

3.1.1 Template Matching Using NCC

Template matching is a technique for identifying regions in an image that match a template [3], [7]. Figure 3.1 shows a template image $T(u, v)$ and an image $I(u, v)$ on which a template match is found. The dimensions of $I(u, v)$ are larger than those of $T(u, v)$. A window with the same dimensions as $T(u, v)$ is moved to all possible locations in $I(u, v)$. At each location, a similarity measure is calculated between the template and image patch under the window.

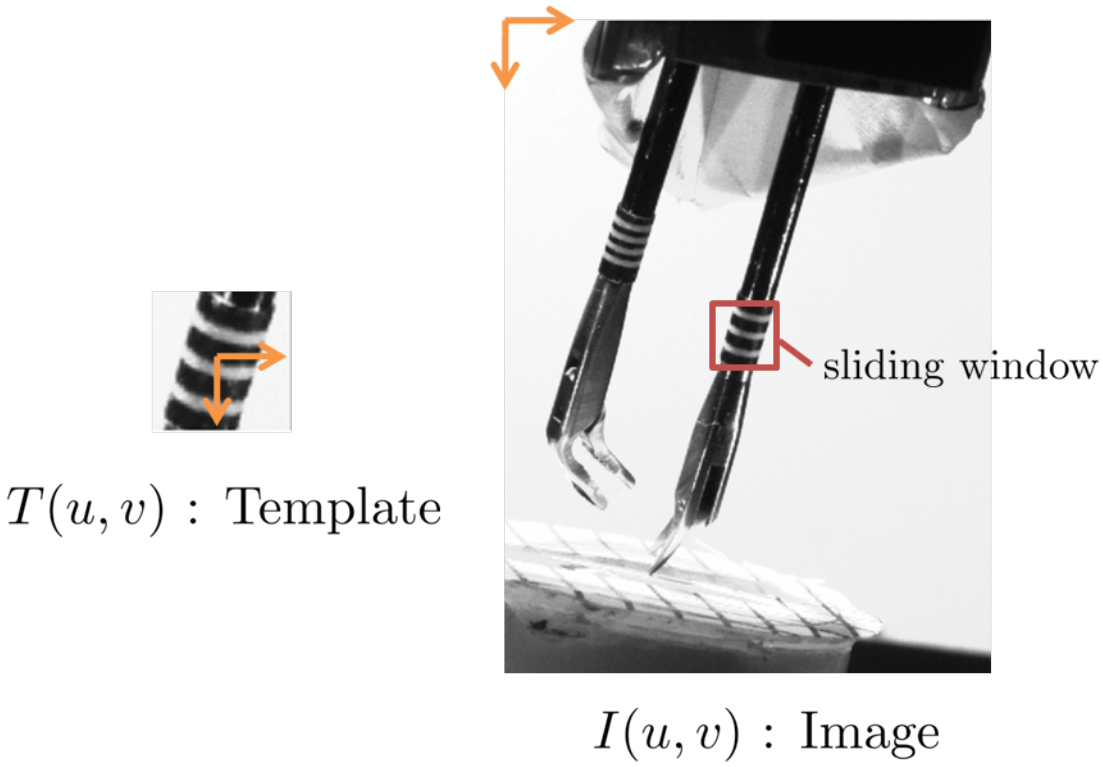

Figure 3.1 Template Matching Overview. A sliding window is used to search for a region in the image that matches the template

This similarity measure is used to find template matches in the image. For example, if one template-like region is expected in the image, the window with the highest similarity score is taken as the location of the template match.

A common measure of similarity is the normalized cross-correlation (NCC) coefficient. Its value can be computed at all possible window positions by calculating the normalized cross-correlation function between the template and the image given by

$$\gamma(a,b) = \frac{\sum_{u,v}\Big(I(u,v) - \bar{I}_{a,b}\Big)\Big(T(u-a,y-b) - \bar{T}\Big)}{\Bigg(\sum_{u,v}\Big(I(u,v) - \bar{I}_{a,b}\Big)^2 \sum_{u,v}\Big(T(u-a,v-b) - \bar{T}\Big)^2\Bigg)^{\frac{1}{2}}}. \tag{3.1}$$

The NCC coefficient for a window centered at (a,b) is given by $\gamma(a,b)$. All summations are taken over the points (u,v) within the window centered on (a,b). The window has the same dimensions as the template and the mean of the image region under the window is given by $\bar{I}_{a,b}$, the mean of the template is given by $\bar{T}$. The numerator of Equation (3.1) is the convolution between the template with its mean removed and the image with its mean subtracted. It is commonly implemented using the Fast Fourier Transform. Normalization is provided by the denominator term, which restricts the values of $\gamma(a,b)$ to $[-1,1]$. The magnitude of the value indicates the degree of similarity. By normalizing the cross-correlation term, the matching process is invariant to scaling in intensity. For example, let the NCC coefficient between the template and an image patch be 0.7. If the pixels in the image patch are multiplied by 0.5, the NCC coefficient will not change. Additionally, due to mean removal, if a constant is added to all the pixels in the image patch the NCC coefficient will remain 0.7. This was useful for tracking phonomicrosurgery instruments as non-uniform lighting and shading (due to one instrument blocking light from the other) caused intensity variations on regions of the instruments being tracked.

The template in Figure 3.1 consists of two objects: a binary marker placed on the cylindrical rod portion of a phonomicrosurgery instrument and the white background behind the surgical instrument. The background region can be excluded from the NCC coefficient calculation by incorporating a binary weighting mask, $w(u,v)$, seen in Figure 3.2. White pixels correspond

$T(u,v)$: Template $\quad w(u,v)$: Binary Weighting

Figure 3.2 Template and Binary Weighting Mask

to indicies of $w(u,v)$ equal to one, while black pixels are indices equal to zero. The template mean and image window mean are computed with weighting the mask as

$$\bar{T} = \frac{\sum_{u,v} w(u,v)T(u,v)}{\sum_{u,v} w(u,v)} \tag{3.2}$$

and

$$\bar{I}_{a,b} = \frac{\sum_{u,v} w(u-a,v-b)I(u,v)}{\sum_{u,v} w(u-a,v-b)}. \tag{3.3}$$

The weighted normalized cross-correlation function is written as

$$\gamma_w(a,b) = \frac{\sum_{u,v} w(u-a,v-b)\Big(I(u,v)-\bar{I}_{a,b}\Big)w(u-a,v-b)\Big(T(u-a,y-b)-\bar{T}\Big)}{\left(\sum_{u,v}\Big(w(u-a,v-b)(I(u,v)-\bar{I}_{a,b})\Big)^2 \sum_{u,v}\Big(w(u-a,v-b)(T(u-a,v-b)-\bar{T})\Big)^2\right)^{\frac{1}{2}}} \tag{3.4}$$

where summations are taken over the points (u,v) within the window centered at (a,b). Let the $\star$ operator represent cross-correlation such that the cross-correlation between two images $J(u,v)$ and $K(u,v)$ is given by

$$(J \star K)(a,b) = \sum_{u,v} J(u,v)K(u-a,v-b). \tag{3.5}$$

Denoting $T'_w(u,v) = w(u,v)(T(u,v)-\bar{T})$ and $T'_{w^2}(u,v) = w^2(u,v)(T(u,v)-\bar{T})$, the weighted NCC in Equation (3.4) can be written in terms of cross-correlations and summations

$$\gamma_w(a,b) = \frac{(I \star T'_{w^2})(a,b) - \frac{\sum_{u,v} T'_{w^2}(u,v)}{\sum_{u,v} w(u,v)}}{\left((I^2 \star w)(a,b) - \frac{2}{\sum_{u,v} w(u,v)}(I \star w)(a,b)(I \star w^2)(a,b) + \right.} \tag{3.6}$$

$$\left.\left(\frac{1}{\sum_{u,v} w(u,v)}(I \star w)(a,b)\right)^2 \sum_{u,v} w^2(u,v) \sum_{u,v} T'_{w^2}(u,v)\right)^{\frac{1}{2}}$$

This function is computed on a pixel grid and provides the template location at a pixel resolution. If one template-like region is expected in the image, the global maxima of $\gamma_w(a,b)$ gives its location. A sub-pixel estimate of the template location can be made by fitting points near the global maxima to a continuous model. Assuming a coordinate system centered on the global maxima of $\gamma_w(a,b)$, a quadratic model of the function local to the maxima is given by

$$C(x,y) = a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 xy + a_5 y^2 \tag{3.7}$$

where the peak location is given by

$$x_{peak} = \frac{2a_1 a_5 - a_2 a_4}{a_4^2 - 4a_3 a_5} \tag{3.8}$$

$$y_{peak} = \frac{2a_2 a_3 - a_1 a_4}{a_4^2 - 4a_3 a_5}. \tag{3.9}$$

The model in Equation (3.7) can be fit using the global maxima and its eight neighboring pixels of the weighted NCC. After a model has been fit, the sub-pixel location of the template is given by the peak in Equations (3.8) and (3.9).

3.1.2 Background Subtraction

Background subtraction is a technqiue for extracting pixels of a foreground object in an image. Let $I(u,v)$ be an image of an object in a background and $B(u,v)$ a model of the background. The background subtraction image is given by

$$S(u,v) = I(u,v) - B(u,v). \tag{3.10}$$

Foreground pixels can be extracting by thresholding this image as

$$\begin{array}{ll} |S(u,v)| \geq \tau & \text{Foreground} \\ |S(u,v)| < \tau & \text{Background} \end{array}. \tag{3.11}$$

In a video, a background model can be used to extract foreground object pixels in every frame. If the background is static, a single image of the background without foreground objects can be used as the background model.

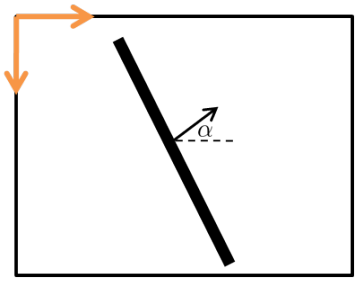

Figure 3.3 Image Gradient at an Edge Pixel. The gradient is oriented normal to the direction of the edge.

3.1.3 Edge Detection Using The Gradient Operator

Let $f(x, y)$ be a scalar valued continuous function. The gradient operator [3] applied to f is

$$\nabla f = grad(f) = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \end{bmatrix}^T . \tag{3.12}$$

It can be thought of as a vector aligned with the maximum rate of change of $f(x, y)$. The magnitude, $M(x, y)$, and the orientation, $\alpha(x, y)$, of this vector are given by the following equations

$$M(x, y) = mag(\nabla f) = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} \tag{3.13}$$

and

$$\alpha(x, y) = tan^{-1}\left(\frac{\frac{\partial f}{\partial y}}{\frac{\partial f}{\partial x}}\right) . \tag{3.14}$$

In a digital image, $I(u, v)$, the gradient is used to determine the edge strength and orientation of a pixel. Figure 3.3 is an illustration of the gradient in an image of a black line with a white background. The vector in the illustration represents the gradient at a pixel in the image and is oriented normal to the edge the pixel lies on. To calculate the gradient of a discrete image, an approximation of the partial derivatives in Equation (3.12) must be used. This can be done with convolution. Let $*$ represent convolution. The Sobel operator is given by the following two equations

$$G_x(u,v) = \begin{bmatrix} -\frac{1}{8} & 0 & \frac{1}{8} \\ -\frac{1}{4} & 0 & \frac{1}{4} \\ -\frac{1}{8} & 0 & \frac{1}{8} \end{bmatrix} * I(u,v) \tag{3.15}$$

and

$$G_y(u,v) = \begin{bmatrix} -\frac{1}{8} & -\frac{1}{4} & -\frac{1}{8} \\ 0 & 0 & 0 \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \end{bmatrix} * I(u,v). \tag{3.16}$$

An approximation of the gradient using the Sobel operator is given by

$$G(u,v) = \begin{bmatrix} G_x(u,v) & G_y(u,v) \end{bmatrix}^T. \tag{3.17}$$

The magnitude and orientation of the gradient are given by

$$M(u,v) = \sqrt{G_x(u,v)^2 + G_y(u,v)^2} \tag{3.18}$$

$$\alpha(x,y) = tan^{-1}\left(\frac{G_y(u,v)}{G_x(u,v)}\right). \tag{3.19}$$

The gradient is commonly used for identifying pixels in an image that belong to edges. This process is known as edge detection. Using criteria based on the gradient, every pixel in the image is labeled as an edge or non-edge pixel. This can be treated as a binary labeling of every pixel in the image as a 1 (edge) or 0 (non-edge). Global magnitude thresholding is one technique for doing this based on thresholding the gradient magnitude, $M(u,v)$. Let $E(u,v)$ be a binary valued image that is the result of applying edge detection on an image $I(u,v)$. The following equation

$$E(u,v) = \begin{cases} 1 & |M(u,v)| > \tau \\ 0 & |M(u,v)| \leq \tau \end{cases} \tag{3.20}$$

gives an expression for global magnitude thresholding, where τ is the global threshold. Thresholding can also be applied to the gradient orientation. For example, vertical edges can be detected as

$$E(u,v) = \begin{cases} 1 & |\alpha(u,v)| > 45 \\ 0 & |\alpha(u,v)| \leq 45 \end{cases}. \tag{3.21}$$

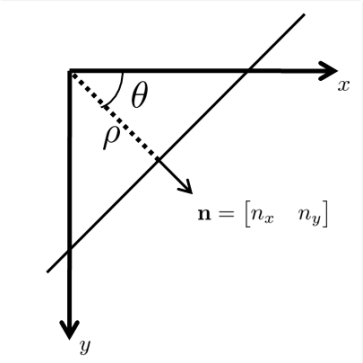

Figure 3.4 Line parameterized as (ρ, θ).

Non-maxima suppression is a technique used to thin connected groups of detected edge pixels [3]. The technique relabels all detected edge pixels that are not local maxima of the gradient magnitude as non-edge pixels. Figure 3.5 gives an illustration of this in which the local neighborhood of a pixel is defined as its left and right neighbor. Part (a) of the figure is an image of a vertical stripe. Part (b) shows a horizontal scanline of the image's gradient magnitude. A red dotted line indicates a global threshold. Near the edges of the stripe, multiple connected pixels will be labeled as edges. Part (c) shows the labeling after non-maxima suppression, in which there are no connected sets of edge pixels.

3.1.4 Line Representation in Images

A line in $\mathbb{R}^2$ can be represented as a point on the projective plane $\mathbb{P}^2 = \mathbb{R}^3 - (0, 0, 0)$ [11]. It is defined by a homogeneous vector $\tilde{\mathbf{l}} = \begin{bmatrix} a & b & c \end{bmatrix}^T$. Points on the line (x, y) in $\mathbb{R}^2$ satisfy the line equation

$$s \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = 0 \tag{3.22}$$

where s is an arbitrary non-zero scale factor. Figure 3.4 illustrates the (ρ, θ) parameterization of a 2D line. The angle between the line's normal vector $\mathbf{n}$ and the x-axis is given by θ. The parameter ρ is the length of a line connected between the origin and the line being parameterized

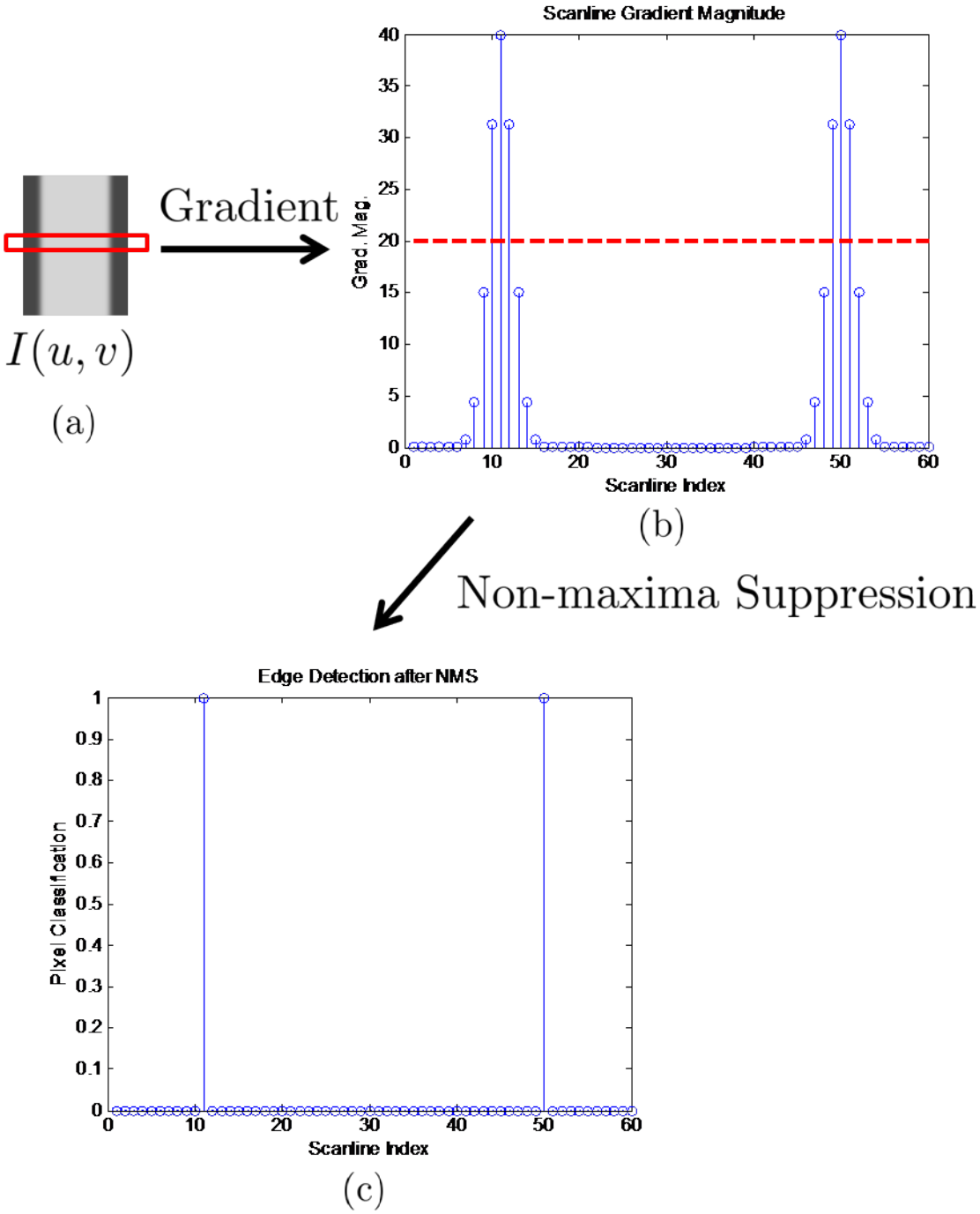

Figure 3.5 Non-Maxima Suppression Example. (a) Image of a vertical stripe (b) Gradient magnitude of a horizontal scanline with global threshold (c) Edge detection after non-maxima suppression

aligned with the normal vector. A point (x, y) on this line in $\mathbb{R}^2$ satisfies the equation

$$\rho = xcos(\theta) + ysin(\theta). \tag{3.23}$$

Note, ρ can be valued positive or negative. It is common to restrict θ to $[-90, 90]$ degrees. In this case, lines that would be parameterized with a positive ρ value and a $\theta > 90$ or $\theta < -90$ are parameterized with a negative ρ value and a θ value within the range $[-90, 90]$ degrees. An equivalent (up to a scale factor difference) homogeneous line representation can be written in terms of these parameters as

$$\begin{bmatrix} a & b & c \end{bmatrix}^T = s \begin{bmatrix} n_x & n_y & -\rho \end{bmatrix}^T = s \begin{bmatrix} cos(\theta) & sin(\theta) & \rho \end{bmatrix}^T \tag{3.24}$$

where s is an arbitrary non-zero scale factor. The orthogonal distance $d_{\rho,\theta}(x, y)$ between a point (x, y) and a line can be computed using

$$d_{\rho,\theta}(x, y) = | \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} cos(\theta) \\ sin(\theta) \\ -\rho \end{bmatrix} |. \tag{3.25}$$

3.1.5 Hough Transform

The Hough Transform is a technique for detecting parametrized shapes in an image [9]. Based on a set of candidate pixels, votes are cast for possible shape parameterizations. The parameterization with the most votes is taken as the detected shape. This voting scheme makes the detection robust to outliers and missing pixels in the candidate set. This subsection describes the use of the Hough Transform for detecting lines with (ρ, θ) parameterization.

Given a set of candidate line pixel locations, the Hough Transform computes votes for possible line parameterizations. An accumulator is a matrix used to count votes. It is a discretization of the space of possible lines that can appear in the image. Assume lines with $\rho = [-100, 100]$ and $\theta = [-90, 90]$ are expected. Let the ρ resolution of the accumulator be 1 and the θ resolution 1. Then the accumulator matrix has dimensions (181×201) where the rows correspond to θ and the columns ρ. Each index of this matrix is a bin for line parameterization votes. For every candidate pixel, votes are cast in the accumulator for possible lines

that the pixel belongs to. At every value of θ in the accumulator, Equation (3.23) is used to calculate the corresponding ρ. A vote is added to the accumulator bin nearest to (ρ, θ). This process is repeated for every candidate pixel. If one line is expected, the bin with the most votes can be used to determine its parameterization. Thresholding on the number of votes can be used to identify a set of lines in an image.

3.1.6 Total Least Squares Line Fitting

When using the Hough Transform, the resolution of line parameter estimates are determined by the accumulator's resolution. A higher resolution estimate can be made by fitting a set of candidate pixels to a line model using Total Least Squares. Unlike the Hough Transform, this technique is not robust to outliers. Given a set of candidate pixels $\{x_i, y_i\}_{i=0}^{N-1}$ the solution to the following minimization

$$\min_{n_x, n_y, \rho} \sum_{i=0}^{N-1} (n_x x_i + n_y y_i - \rho)^2 \tag{3.26}$$

are the line parameters $n_x = cos(\theta), n_y = sin(\theta), \rho$ that minimize the sum of squared orthogonal fitting errors. When solving this minimization, the constraint $\sqrt{n_x^2 + n_y^2} = 1$ must be satisfied. Alternatively, the minimization can be reposed [11] as

$$\min_{a,b,c} \sum_{i=0}^{N-1} (ax_i + by_i + c)^2 \quad S.T. \quad \sqrt{a^2 + b^2 + c^2} = 1, \tag{3.27}$$

and the Singular Value Decomposition can be used to solve for the parameters. The solution is within a scale factor of $\begin{bmatrix} cos(\theta) & sin(\theta) & -\rho \end{bmatrix}$. By identifying the scale factor, the line parameters (ρ, θ) can be found. Let s represent the scale factor. Its magnitude is given by

$$|s| = \frac{1}{\sqrt{a^2 + b^2}}. \tag{3.28}$$

After determining magnitude, its sign is chosen such that θ in the equation,

$$\theta = cos^{-1}(\frac{a}{s}) = sin^{-1}(\frac{b}{s}) \tag{3.29}$$

is in the range $[-90, 90]$ degrees.

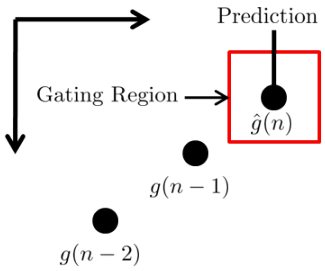

Figure 3.6 Gating Region Defined by Constant Velocity Prediction Model. The search for the feature in frame n is restricted to the gating region defined by the red box.

3.1.7 Prediction Models and Gating

Let $\mathbf{g}(n)$ be a feature being tracked in a video. The elements of this vector are related to the position of the feature. A prediction model of the feature provides a prediction of the future location of the feature based on its current and past locations. Two examples of prediction models are the constant position model

$$\hat{\mathbf{g}}(n+1) = \mathbf{g}(n) \tag{3.30}$$

and the constant velocity model

$$\hat{\mathbf{g}}(n+1) = \mathbf{g}(n) + (\mathbf{g}(n) - \mathbf{g}(n-1)). \tag{3.31}$$

One method for tracking a feature is to search for it over all possible positions in every frame. For example, if a region is being tracked the NCC between a template and the entire frame can be calculated. A search for the position with the maximum normalized cross-correlation coefficient is performed over the entire image. The position of the maximum is taken as the region's position in the current frame. Gating is a restriction applied to this search based on a prediction model for the feature. In this example, the past two region locations can be used to make a prediction for the current region location. The NCC between the template and a gating window centered on the predicted location is calculated. The maximum in the gating window is taken as the region's location. Figure 3.6 illustrates this concept. A prediction of a feature's current position $\hat{\mathbf{g}}(n)$ is made using its past two values. A red box defines

the gating window surrounding the prediction. Gating allows the amount of computation performed at each frame to be reduced, but requires the proper selection of gating window size and prediction model.

3.2 Monocular Video Instrument Tracking Algorithm

This section describes the algorithm used to track a single phonomicrosurgery instrument in a video of a simulated exercise. Tracking consists of finding a set of instrument features in each frame. Figure 3.7 shows these features: a window enclosing a binary maker placed on the cylindrical rod of the instrument, lines along the instrument's visible boundaries, the imaged midline, and a track point given by the intersection of the midline and the vertical midpoint of the window. These features are representative of the instrument's two-dimensional position and orientation. Over an entire video, they give its two-dimensional trajectory. First, the algorithm used to track the window enclosing the binary marker is described. Next, the algorithm used to track the instrument's boundary lines is described. Finally, the entire tracking algorithm incorporating window and line tracking is described.

3.2.1 Instrument Marker Tracking

One of the features tracked by the algorithm is a window containing the image of a passive binary marker placed on the cylindrical rod of the instrument. Figure 3.7 shows a window enclosing a marker attached to a phonomicrosurgery instrument. The goal of tracking the marker is to locate the same region of an instrument in every video frame. The marker is printed on a paper label with adhesive. A paper marker was selected over the actual instrument appearance due to reflectance and texture properties. The surface of the phonomicrosrugery instrument consists of specular reflections and low texture regions. Therefore, regions of the instrument are not distinctive and difficult to track. Black and white bars running perpendicular to instrument edges on the passive marker provide a distinctive appearance that can easily be identified along the length of the instrument. Additionally, when viewing multiple instruments variation in the thickness of the stripes can be utilized to distinguish between the two instruments.

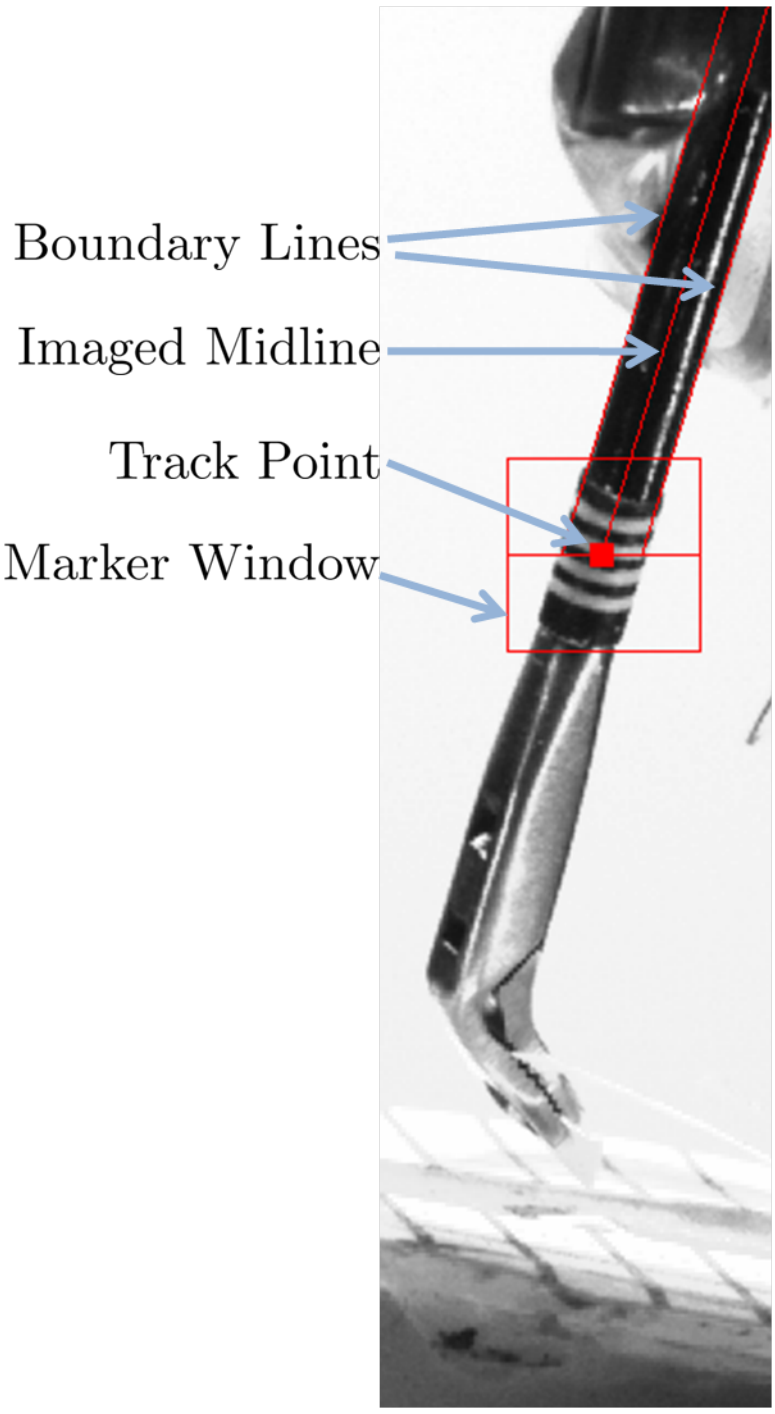

Figure 3.7 Instrument Features Detected Using Tracking Algorithm.

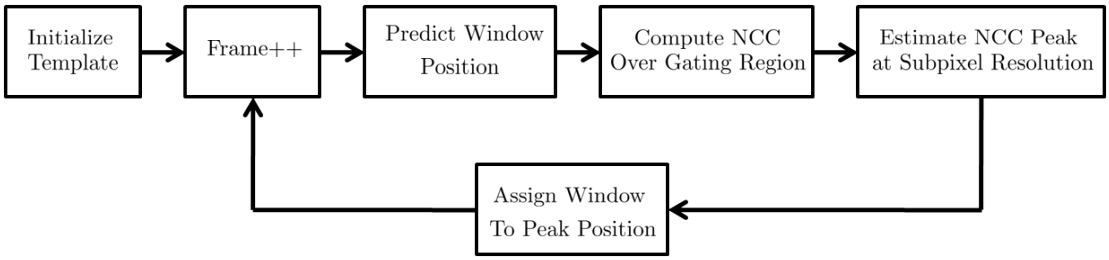

Figure 3.8 Marker Window Tracking Algorithm Block Diagram.

The stripe pattern of the marker also makes it invariant to rotation about the instrument's axis. When the instrument is rotated about its axis, the region of the instrument with the marker does not change significantly in appearance.

In terms of feature tracking, a window centered on the image of the marker must be found in every frame. Figure 3.8 is a block diagram of the algorithm used to do this. Let $\mathbf{m}(n) = \begin{bmatrix} u & v \end{bmatrix}$ be a signal representing the center of the window (u, v) at frame n. In the first tracking frame a window enclosing the marker is manually selected. This window defines a template, $T(u, v)$. A binary mask $w(u, v)$ within $T(u, v)$ is selected to include the marker but exclude background present in the window. A template matching scheme using these two components is used to track the marker region. In each frame, the location of the window is predicted using the constant velocity model in Equation (3.31). A gating region is centered on the predicted position. The size of this region is controlled by a user defined parameter. The weighted NCC between the image within the gating region and the template is calculated. A pixel resolution estimate of the marker window's position is given by the maximum of the weighted NCC. A subpixel estimate of the window's position is calculated using quadratic subpixel peak fitting.

3.2.2 Boundary Line Tracking

After the marker window position has been detected, two lines at the instrument's visible boundaries are found. They are seen in Figure 3.7. Let (ρ_L, θ_L) and (ρ_R, θ_R) denote the parameterization of these lines. An estimate of the instrument's imaged midline $(\hat{\rho}_{mid}, \hat{\theta}_{mid})$ is calculated from these lines using the following two equations

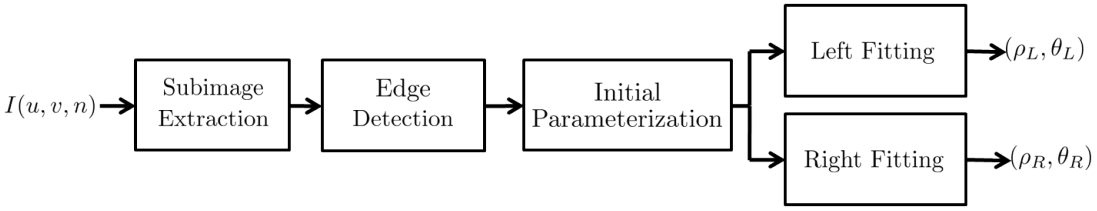

Figure 3.9 Boundary Line Pair Detection Block Diagram)

$$\hat{\rho}_{mid} = \frac{\rho_L + \rho_R}{2} \tag{3.32}$$

and

$$\hat{\theta}_{mid} = \frac{\theta_L + \theta_R}{2}. \tag{3.33}$$

This requires that the θ parameter is restricted to the range $[-90, 90]$ degrees. In every frame, a point on the imaged midline called the instrument's track point is found. It is seen in Figure 3.7. Its position is given by the intersection of the imaged midline and the vertical midpoint of the marker window.

Boundary line tracking consists of four steps: subimage extraction, edge detection, initial line pair parameterization, and final line fitting. Figure 3.9 is a block diagram showing the interconnection between these steps. First, a subimage is extracted from the current frame. Next, a set of left-right instrument edge pixels are extracted from this subimage. Then, an initial boundary line pair parameterization is found using the Hough Transform. Finally, a left and right boundary line parameterization are calculated using a total least squares fitting procedure.

The goal of subimage extraction is to extract the region of the frame containing the instrument's cylindrical rod. This reduces the amount of computation and number of outliers detected in the following edge detection step. First, the shape of the subimage is defined using the feature values from the previous frame. The displacement of the marker window between the previous and current frame is used to propagate the subimage into the current frame. Figure 3.10 illustrates this concept, where $\mathbf{m}(n)$ represents the position of the marker window. The red line in the figure is the detected instrument midline in the previous frame. The dotted parallelogram is the subimage. Two of its sides are parallel to the image's x-axis. Its other two

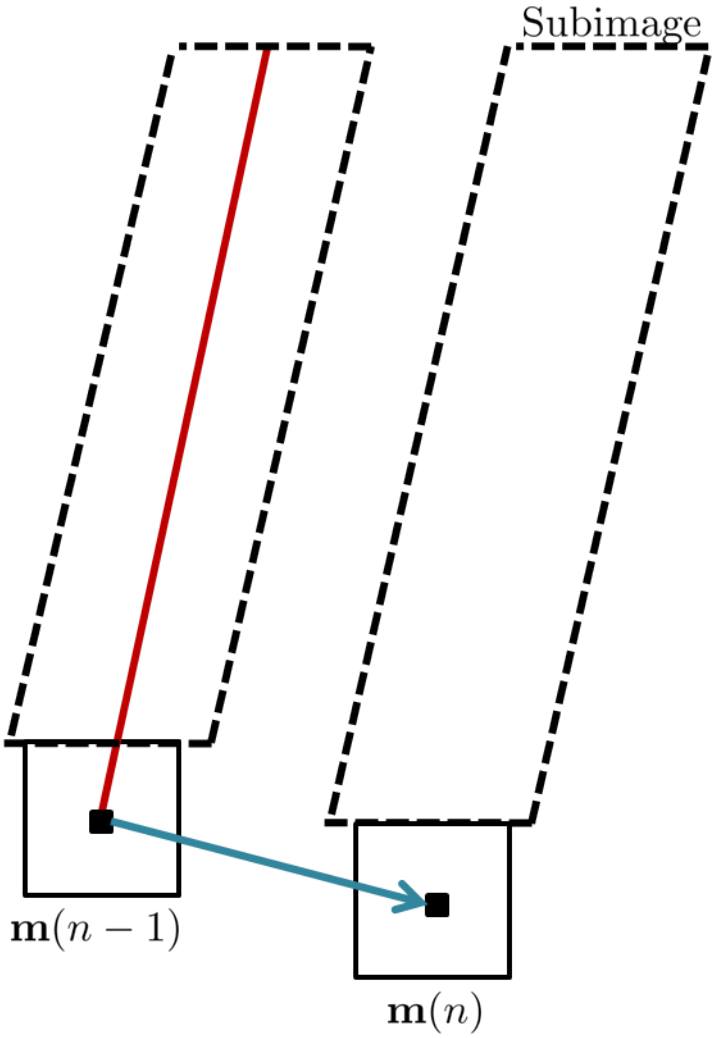

Figure 3.10 Subimage Propogation. A subimage is first defined using the marker window position $\mathbf{m}(n-1)$ and midline estimate (red line) in the previous frame. This subimage is displaced by the marker window displacement $(\mathbf{m}(n) - \mathbf{m}(n-1))$. This defines the subimage in the current frame at which boundary line tracking is being performed.

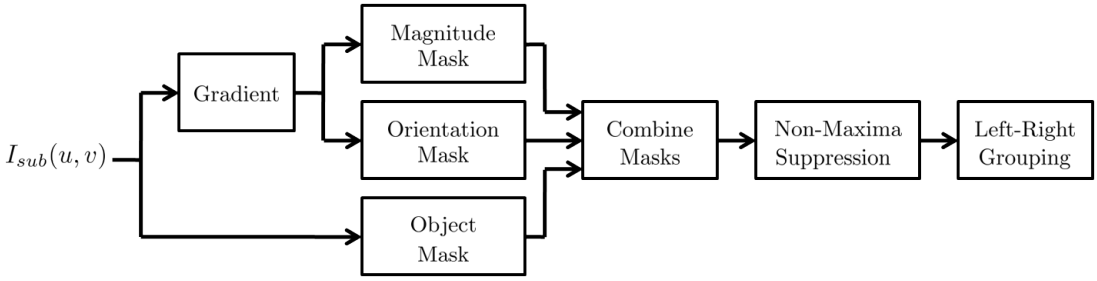

Figure 3.11 Edge Detection Block Diagram

sides are oriented parallel to the instrument's imaged midline in the previous frame. Vertically, it extends from the top of the marker window to to the top of the image. Its horizontal width is controlled by a user-defined parameter. The image within the subimage region is used in the edge detection step.

The edge detection step in Figure 3.9 identifies a set of pixels along the left and right boundary of the instrument. Figure 3.11 is a block diagram of the edge detection algorithm. Edges are detected in a subimage $I_{sub}(u,v)$ of the instrument's cylindrical rod region. First, a set of binary masks : $I_{obj}(u,v), I_{mag}(u,v), I_{orient}(u,v)$ are computed. The gradient of $I_{sub}(u,v)$ is used to compute a gradient magnitude $I_{mag}(u,v)$ and orientation $I_{orient}(u,v)$ mask. Pixel's with a gradient magnitude above a user defined threshold are labeled as ones in $I_{mag}(u,v)$. If a pixel's orientation indicates a vertical edge, it is labeled as one in $I_{orient}(u,v)$. Background subtraction is used to compute the object mask $I_{obj}(u,v)$. Let $B_{sub}(u,v)$ be a background subimage of the surgical setup without any instrument's present. Pixel's in $I_{obj}(u,v)$ are labeled according to

$$I_{obj}(u,v) = \begin{cases} 1 & B_{sub}(u,v) - I_{sub}(u,v) > \tau_{back} \\ 0 & B_{sub}(u,v) - I_{sub}(u,v) \leq \tau_{back} \end{cases} . \tag{3.34}$$

This mask consists of labeled pixels belonging to the surgical instrument. The three masks are combined using a logical AND operation. The resulting mask consists of pixels belonging to vertical instrument edges. Non-Maxima suppression is used to thin this edge set. If a pixel is not a local maximum with respect to its left and right neighbor, it is suppressed. After Non-Maxima suppression, the final set of edge pixels is found. This set is grouped into a left and

right edge set. Gradient orientation is used to perform the grouping. The lighting of the setup is controlled such that the instrument appears dark in a light background. Let $G_x(u, v)$ be the x component of the gradient calculated with the Sobel operator. Pixels are assigned as left and right edges according to

$$\begin{array}{ll} G_x(u,v) < 0 & \text{Left Edge} \\ G_x(u,v) \geq 0 & \text{Right Edge} \end{array} . \tag{3.35}$$

The Hough Transform is used to determine an initial parameterization of the instrument's boundary lines. Let $(u_L^k, v_L^k)\}_{k=1}^{N_L}$ and $\{(u_R^m, v_R^m)\}_{m=1}^{N_R}$ be the set of left and right pixels found in the edge detection step. An accumulator for each set of pixels H_L and H_R is found using the Hough Transform

$$\begin{array}{l} \{(u_L^k, v_L^k)\}_{k=1}^{N_L} \rightarrow H_L(\begin{bmatrix} \rho \\ \theta \end{bmatrix}) \\ \{(u_R^m, v_R^m)\}_{m=1}^{N_R} \rightarrow H_R(\begin{bmatrix} \rho \\ \theta \end{bmatrix}) \end{array} . \tag{3.36}$$

For every left line parameterization in H_L with more than zero votes, a matching right line parameterization is found. These two parameterizations form a boundary line parameterization pair. The right parameterization in this pair is found by a searching a subregion of the right accumulator. Figure 3.12 illustrates the subregion with an orange rectangle. Let (ρ_L, θ_L) be a left line parameterization. The search region in the right accumulator is defined by the ranges $[\rho_L + \Delta\rho_{min}, \rho_L + \Delta\rho_{max}]$ and $[\theta_L + \Delta\theta_{min}, \theta_L + \Delta\theta_{max}]$. The parameters $\Delta\rho_{min}$ and $\Delta\rho_{max}$ are chosen based on the expected instrument width. If the lines were expected to be perfectly parallel, parameters $\Delta\theta_{min}$ and $\Delta\theta_{max}$ would be set to zero. Due to depth variation along the length of the instrument, imaged boundaries are not perfectly parallel. The parameters $\Delta\theta_{min}$ and $\Delta\theta_{max}$ are chosen such that line pairs near parallel are found. Let

$$\mathbf{M}(\begin{bmatrix} \rho_L \\ \theta_L \end{bmatrix}) = \begin{bmatrix} \rho_R \\ \theta_R \end{bmatrix} \tag{3.37}$$

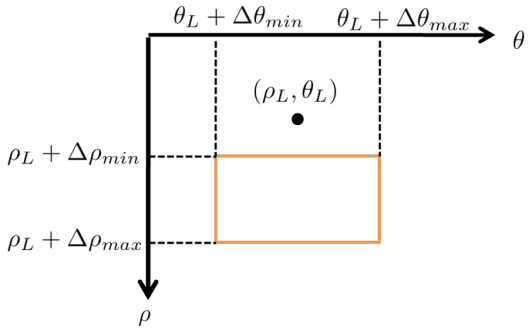

Figure 3.12 Right Accumulator Search Region for Initial Boundary Pair Detection. A search region in the right accumulator is given by the orange rectangle. It is defined with respect to the left line parameterization (ρ_L, θ_L).

represent a mapping between a left line parameterization and the right line parameterization with which it forms a boundary line pair. A line-pair accumulator is given by

$$H_{L,R}(\begin{bmatrix}\rho_L \\ \theta_L\end{bmatrix}) = H_L(\begin{bmatrix}\rho_L \\ \theta_L\end{bmatrix}) + H_R\left(\mathbf{M}(\begin{bmatrix}\rho_L \\ \theta_L\end{bmatrix})\right). \tag{3.38}$$

The initial boundary line parameterization pair is given by the pair in this accumulator with the most votes.

The Hough Transform is used to robustly find an initial boundary line-pair parameterization of a surgical instrument. The resolution of the Hough accumulator limits the resolution of this estimate. The final step in the boundary line tracking process is to fit a higher resolution estimate of the boundary line parameters using Total Least Squares. The left and right boundary lines are fit independently. Fitting consists of two steps. First, a subset of inliers in the set of edge pixels is found using the initial boundary line parameterization. Second, a line is fit to the set of inliers using Total Least Squares. Let $(u_L^k, v_L^k)\}_{k=1}^{N_L}$ be the left edge pixels and (ρ'_L, θ'_L) the initial boundary line parameterization found using the Hough Transform. The inlier set of edge pixels is given by

$$\{(u,v) \in \{(u_L^k, v_L^k)\}_{k=1}^{N_L} | d_{\rho_L,\theta_L}(u,v) \leq \tau_{inlier}\} \tag{3.39}$$

where $d_{\rho_L,\theta_L}(u,v)$ is the orthogonal distance between a point (u,v) and a line parameterized by (ρ'_L, θ'_L) and τ_{inlier} is a threshold used to determine inlier pixels. The set of inliers is used

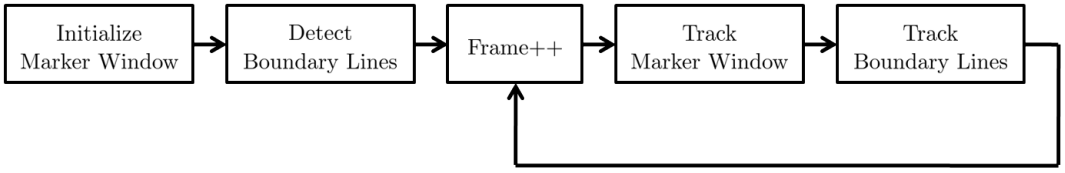

Figure 3.13 Full Instrument Tracking Algorithm.

to make a final line fitting (ρ_L, θ_L) using Total Least Squares. For a single instrument, this is performed for the left and right boundary independently.

3.2.3 Instrument Tracking Algorithm Overview

Figure 3.13 is a high-level block diagram of the instrument tracking algorithm. In the marker window tracking step, the position of a window $\mathbf{m}(n)$ containing a binary marker placed on the instrument is found. In the boundary line tracking step, a parameterization of the instrument's visible left $(\rho_L(n), \theta_L(n))$ and right $(\rho_R(n), \theta_R(n))$ boundary are found. These boundary lines are used to estimate the instrument's imaged midline using equations (3.32) and (3.33). The intersection of the midline and the vertical midpoint of the marker window defines the instrument's track point.

Prior to tracking, the marker window is initialized manually. This consists of setting the window's initial position and size, a template used for tracking, and a weighting mask used during tracking. After window initialization, the first pair of boundary lines are detected. The boundary line tracking algorithm utilizes the previous midline estimate and window displacement to first extract a subimage. At the first frame this data is not available. Instead, the shape of the weighting mask is used define the position and orientation of the subimage.

3.3 Algorithm Confidence

The tracking algorithm will incorrectly identify instrument features in frames containing significant motion blur and/or occlusion. Figure 3.14 shows examples. The features detected in these frames can deviate significantly from the actual instrument features. If tracking is continued using these values, it is likely that the algorithm will continue to incorrectly track the

features. To prevent this from happening, the tracking algorithm's confidence is evaluated after each frame. First, a set of confidence parameters are calculated. These parameters are input into a binary classifier. This classifier determines if the algorithm is confident or underconfident about the feature values it has found.

3.3.1 Cofindence Parameters

After the instrument features are found using the tracking algorithm, a set of confidence parameters are calculated. These parameters were selected based on the following idea : certain characteristics of the imaged instrument should vary smoothly as the instrument moves. It is expected that the width and orientation of the instrument should vary smoothly. Let $(\rho_L(n), \theta_L(n))$ and $(\rho_R(n), \theta_R(n))$ be the boundary line parameterizations found by the tracking algorithm and n a variable representing frame number. The first confidence parameter is given by

$$p_1(n) = |\rho_L(n) - \rho_R(n)| \tag{3.40}$$

and is a measure of instrument width. Instrument orientation is given by

$$p_2(n) = \frac{\theta_L(n) + \theta_R(n)}{2}. \tag{3.41}$$

As the instrument moves, the image of the marker region should vary smoothly. The third parameter $p_3(n)$ is the NCC coefficient found using template matching at frame n. A set of inlier points are used to make the final boundary line fitting. Let $i_L(n)$ and $i_R(n)$ be the number of left and right inlier points used for this fitting in frame n. Let, $y_{mark}(n)$ be the y position of the upper left hand corner of the marker window in frame n. These quantities are used to compute the final two confidence parameters

$$p_4(n) = y_{mark}(n) - i_L(n) \tag{3.42}$$

and

$$p_5(n) = y_{mark}(n) - i_R(n). \tag{3.43}$$

These two parameters were selected because the number of inlier fitting points should vary smoothly as the position of the instrument changes. If the instrument moves in the y direction,

Figure 3.14 Frames Containing Motion Blur (left) and Occlusion (right)

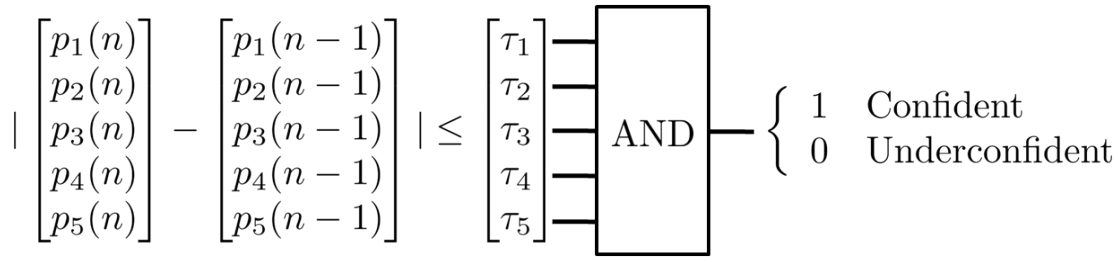

Figure 3.15 Algorithm Confidence Classifier Structure.

the number of available points for line fitting increases or decreases. To prevent this movement from appearing like a tracking error, the y position of the marker window was incorporated into the two parameters.

3.3.2 Confidence Classifier

The classifier used to identify algorithm confidence is illustrated in Figure 3.15. At frame n, the smoothness of parameter p_i is evaluated by computing $|p_i(n) - p_i(n-1)|$. This smoothness value is compared to a smoothness threshold τ_i. If all of the parameters are below their respective threshold, the algorithm is confident in the features it has found. If any of the parameters exceed its respective threshold, the algorithm is underconfident.

3.3.3 User Guided Tracking

The instrument tracking algorithm is augmented with the confidence classifier to form a user-guided instrument tracking scheme. In this scheme, instrument tracking is semi-automated. After initialization, user intervention is needed only when the algorithm is underconfident. Figure 3.16 is a flow diagram of the scheme. First, the marker window is initialized. Then, features are found using the tracking algorithm. Next, algorithm confidence is assessed using the confidence classifier. If the algorithm is confident, tracking is performed on the next frame without user intervention. If the algorithm is underconfident, user intervention is needed. The features found by the tracking algorithm are graphically presented to the user and he/she must verify if the algorithm has correctly found the features. If it has, the user instructs the system to continue

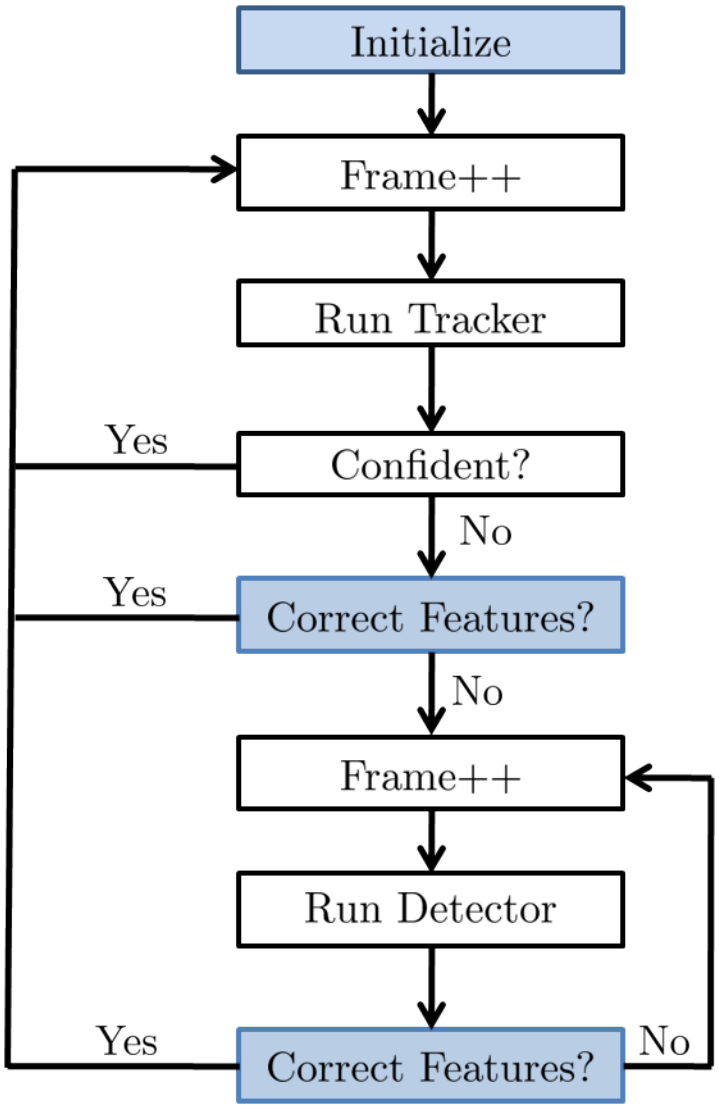

Figure 3.16 User Guided Tracking Scheme. White block indicates automated step. Blue block indicates step requiring user intervention.

tracking. If not, the feature is labeled as an error frame and the feature data for that frame is ignored. Next, the instrument features are detected in the next frame. The user must verify that the detected features are correct. If they are not, the frame is labeled as an error frame and this process is repeated. Once a frame with correctly detected features has been verified, features in the next frame are found using the automated tracking algorithm.

3.3.4 Selecting Confidence Classifier Thresholds

The confidence classifier illustrated in Figure 3.15 uses a set of thresholds to perform confidence classification. These thresholds were set empirically using a training data set. The data set consisted of a left and right view video of a foreceps and scissors instrument being used to perform a cutting exercise. Instrument features on the scissors were tracked without the confidence classifier. Instead, the features found in every frame were manually verified. This verification consisted of identifying correctly tracked frames, and frames in which tracking could not be performed due to blur or occlusion. The number of correctly tracked frames greatly outnumbered the frames at which tracking could not be performed. The training video had 2000 frames and ten frames contained blur and occlusion events that lead to incorrect tracking. Therefore, the threshold parameters were set based on the feature data from the correctly tracked frames. Classification at the incorrectly tracked frames was used to validate the classifier.

Each smoothness threshold was set individually. Viewing each each threshold as an individual classifier, each threshold was set so that 98% of the correctly tracked frames were classified as confident. The actual classifier was formed by combining the thresholds as seen in Figure 3.15. Performance of the classifer was evaluated using the recall and true negative rate statistics

$$Recall = \frac{tp}{tp + fn} \tag{3.44}$$

$$TNR = \frac{tn}{tn + fp} \tag{3.45}$$

where tp is number of true positives, fn is the number of false negatives, tn is the number of true negatives, and fp is the number of false positives. Ideally, correctly tracked frames would be classified as confident and incorrectly tracked frames would be classified as underconfident. True positive corresponds to correctly tracked frames that were classified confident. True negative corresponds to incorrectly tracked frames that were classified underconfident. False positive corresponds to incorrectly tracked frames that were classified confident. False negative corresponds to correctly tracked frames that were classified underconfident. Recall measures the fraction of correctly tracked frames that were properly classified as confident. When the algorithm is confident about a frame, it continues instrument tracking automatically. A larger recall value means less user intervention when running the tracker. True negative rate is the fraction of incorrectly tracked frames that the algorithm is correctly underconfident about.

Table 3.1 contains information regarding evaluation of the confidence classifier. Rows Recall and TNR refer to the statistic values found for a dataset using the threshold parameters fitted on the training dataset. Rows Correctly Tracked Frames and Incorrectly Tracked Frames refer to the number of each frame type in the respective datasets. The different datasets are given by the columns Training and Test. The training data was generated from a video of two cutting events. Features on scissors used for cutting were tracked. Correctly and incorrectly tracked frames were manually labeled. The same procedure was used for the test dataset. It was generated from a video of six cutting events. A recall of 0.94 and TNR of 1.0 were found for the training dataset. These were deemed appropriate values as the majority of frames could be tracked automatically without missing an incorrectly tracked frame. A recall of 0.97 and TNR of 1.0 were found for the test dataset. This validated the use of the threshold parameters found using the training dataset.

3.4 Chapter Conclusion

This chapter described an algorithm for tracking a phonomicrosurgery instrument in a video of a simulated surgical exercise. In every frame of the video, the algorithm finds a set of features related to the position and orientation of the instrument. In V-PITS (the system developed

	Training	Test
Recall	0.94	0.98
TNR	1.00	1.00
Correctly Tracked Frames	1974	17700
Incorrectly Tracked Frames	10	37

Table 3.1 Confidence Classifier Evaluation Information

in this project), a user-guided scheme is used to perform the tracking. This semi-automated scheme was used because blur and occlusion in a video can cause the tracking algorithm to fail. The user-guided scheme automatically tracks the instrument's features until it encounters a frame it is underconfident about. At this point, the user guides the algorithm until it is once again confident and can continue to run automatically. V-PITS was developed to estimate the 3D trajectory of a surgical instrument from video data. The next chapter describes how this is done using the tracked feature data.

Chapter 4

Instrument Trajectory Estimation Algorithm

This chapter describes the algorithm used within V-PITS to estimate the 3D trajectory of a point on a surgical instrument's midline. Using images from a calibrated stereo camera rig, Chapter 2 described a method for estimating the 3D position of a point on a surgical instrument's midline. Chapter 3 described an algorithm for tracking multiple instrument features in a monocular video. The algorithm in this chapter applies the position estimation method at each video frame using the tracked features. The position estimation method relies on knowledge of the stereo camera rig's calibration parameters. First, the method used to estimate these parameters is described. Then, the trajectory estimation algorithm is described.

4.1 Stereo Camera Calibration

4.1.1 Camera Model

V-PITS uses a stereo camera rig to capture synchronized video from two different viewpoints. In order to incorporate information from both viewpoints, a model of the stereo camera rig is used. This model was described in Chapter 2 Subsections 2.1.3 and 2.1.4. It consists of a set of intrinisic parameters for each camera and a set of extrinisic parameters. The intrinsic parameters model the mapping between a point in the 3D world and its 2D image. The extrinsic parameters describe the spatial relationship between the two cameras.

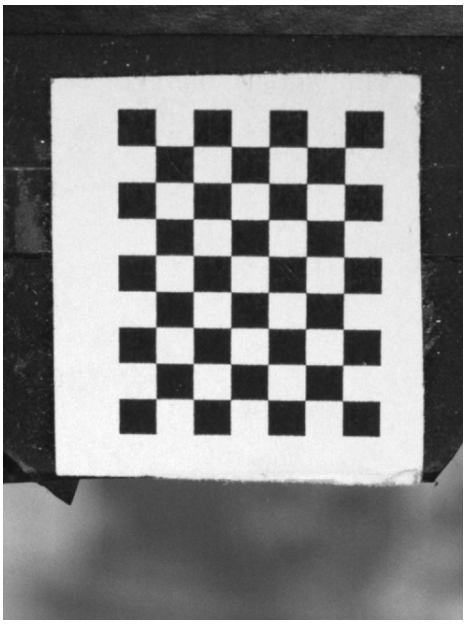

Figure 4.1 Checkerboard Pattern Used for Calibration.

4.1.2 Estimating Calibration Parameters

The Caltech Camera Calibration Toolbox for Matlab [1] is used to estimate camera calibration parameters. First, multiple images of a planar calibration pattern are captured. Then these images are used to estimate each camera's intrinsic parameters independently. Finally, a stereo calibration routine is used to estimate the extrinsic parameters and make a joint estimate of each camera's intrinsic parameters. The calibration pattern is a planar checkerboard as seen in Figure 4.1. The side-length of each square in this pattern is 2.1167 mm. Several images of the pattern at different positions and orientations are taken with the stereo camera rig. The toolbox contains tools to semi-automatically extract corners in an image of the pattern. The corner positions are used by routines within the toolbox to estimate calibration parameters.

4.2 3D Trajectory Estimation

Given videos taken using the stereo camera rig, the 3D trajectory of a phonomicrosurgery instrument refers to the 3D position of a point on the instrument in each frame of the video.

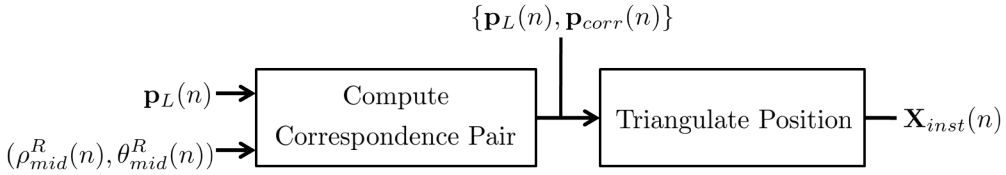

Figure 4.2 3D Trajectory Estimation Scheme.

This is given by

$$\mathbf{X}_{inst}(n) = \begin{bmatrix} X(n) & Y(n) & Z(n) \end{bmatrix}^T \tag{4.1}$$

where n is the frame number. To estimate 3D trajectory, the scheme in Figure 4.2 is used. First, the tracking algorithm in Chapter 3 is run for the same instrument in both videos. This results in a pair of midlines $(\rho^L_{mid}(n), \theta^L_{mid}(n))$,$(\rho^R_{mid}(n), \theta^R_{mid}(n))$ and trackpoints $\mathbf{p}_L(n)$,$\mathbf{p}_R(n)$ for the instrument in the two videos. The trackpoint is a point on the instrument's midline that is tracked over the course of the video. At each frame, a point correspondence pair is found between the two videos using the method described in Chapter 2. The trackpoint in the left camera's video is one point in this correspondence pair. It is used to compute an epipolar line in the right camera's video. The intersection of this epipolar line with the imaged midline in the right camera's video defines the point $\mathbf{p}_{corr}(n)$ in the correspondence point pair. Finally, the 3D position corresponding to the point pair $\{\mathbf{p}_L(n), \mathbf{p}_{corr}(n)\}$ is triangulated using the midpoint algorithm. By performing this process at each video frame, the trajectory of the instrument $\mathbf{X}_{inst}(n)$ is found. Specifically, this trajectory is the 3D position of the left camera's trackpoint at each frame.

This chapter described the algorithm used within V-PITS estimate the 3D trajectory of a phonomicrosurgery instrument. It relies on instrument features tracked using the algorithm in Chapter 3. The method in Chapter 2 is used to form a correspondence point pair and triangulate the 3D position at each video frame. The next chapter describes experiments performed to evaluate characteristics of V-PITS.

Chapter 5

Experimental Characterization

A set of videos were collected in which a surgical instrument-like rod was displaced a known distance. After data collection, the tracking algorithm and instrument trajectory estimation algorithm were run on the videos. In this chapter, the resulting tracked 2D features and 3D instrument trajectory are analyzed to characterize V-PITS. Specifically, static noise and displacement measurement accuracy are characterized. First, the data collection methodology is described. Next, 2D and 3D static noise are characterized. Finally, 3D displacement measurement accuracy is characterized.

5.1 Data Collection Methodology

A set of experimental trials were performed using a mechanical displacement device. A rod with the same outer diameter as a surgical instrument was attached to the mechanical displacement device. Prior to a trial, the rod was positioned in the field of view of both cameras to emulate a surgical instrument. Figure 5.1 contains images from the left and right camera during an experimental trial. During a trial, the mechanical displacement device was used to displace the rod a known distance. Video of the displacement was captured using the stereo camera rig. Between experimental trials the initial position of the rod was adjusted. Three datasets of multiple experimental trials were collected. Within each dataset, the direction of rod displacement was held fixed.

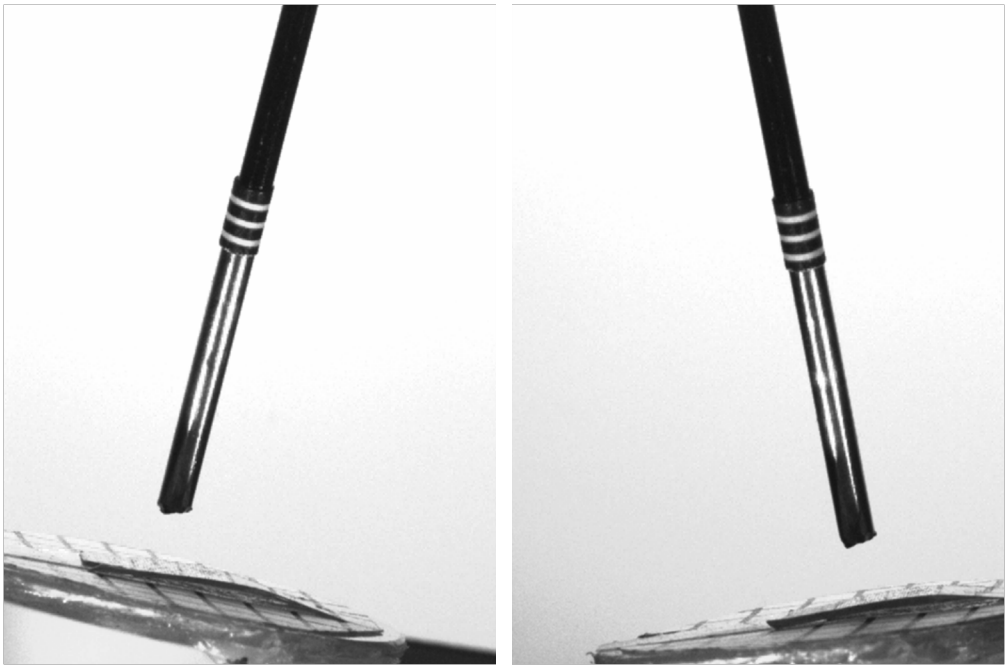

Figure 5.1 Left and Right Camera Frame of Instrument-like Rod.

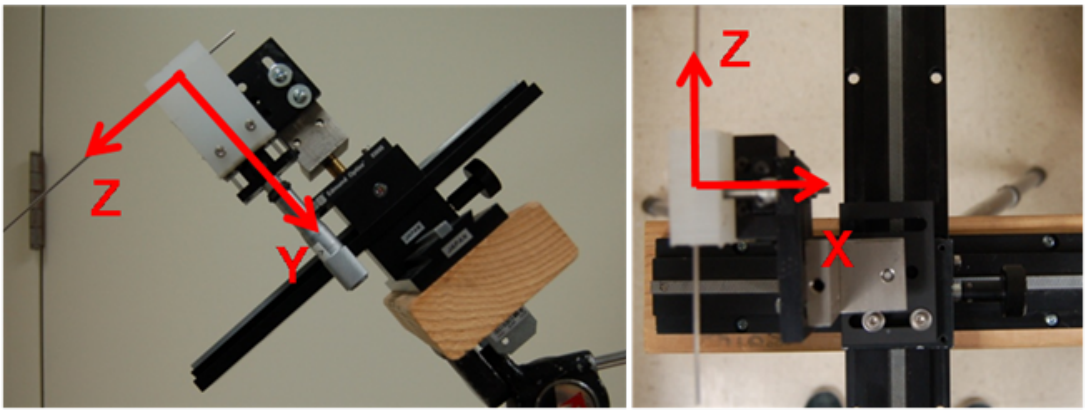

Figure 5.2 Mechanical Displacement Device with Instrument Coordinate Frame Labels. The length of the rod is aligned with the Z axis.

5.1.1 Mechanical Displacement Device

A cylindrical metal rod with 2mm outer diameter was used to simulate a phonomicrosurgery instrument. A marker was attached 15 mm from the end of the rod to mimic the marker location on an actual instrument. The rod was attached to a mechanical displacement device that consists of a tripod, two coarse linear stages, and a precision linear stage. The tripod and coarse linear stages were used to control the position and angle of the rod prior to a trial. The precision stage was used to displace the instrument-like rod 1 mm during an experimental trial. Two images of the displacement device are seen in Figure 5.2. The left hand image is the displacement device viewed from the side and the right hand image is the displacement device viewed from overhead. A coordinate frame associated with the rod is given by red overlays in the figure. The length of the rod runs along the Z axis. This coordinate frame was chosen to be similar to that of a subject manipulating a surgical instrument. Adjustments in instrument depth occur along the Z axis, changes in vertical instrument position occur along the Y axis, and adjustments in horizontal instrument position occur along the X axis.

5.1.2 Experimental Procedure

A single experimental trial consisted of positioning the instrument-like rod and displacing it 1 mm. Video capture started prior to the displacement event and ended ten seconds after the

instrument had been displaced. This was done to capture the instrument over the course of the displacement and after any vibrations due to the movement had dampened out. Three video datasets of experimental trials were collected. In each dataset the direction of displacement was fixed to one of the instrument axes seen in Figure 5.2. Over a single dataset the position of the instrument prior to displacement was varied. This was done to collect trials over the space of possible surgical instrument positions. Between experimental trials the position of the instrument was translated 2-3 mm along one of its axis. Figure 5.3 is a 3D scatter plot of the different instrument starting positions in a single dataset. Each circle corresponds to an individual trial. Here the $X, Y,$ and Z axis refer to the camera frame of the left camera in the stereo rig. By varying the 3D position of the instrument-like rod, the 2D starting position of the instrument's marker was varied. Figure 5.4 is a mosaic of the first frame of multiple videos within a single dataset. Note the variation in the position of the instrument and attached marker.

The tracking algorithm described in Chapter 3 was run on all the videos in each dataset. Figure 5.5 shows the instrument's track point position for a single experimental trial video. The step-like movement is related to the manual stage used to displace the instrument. Using the algorithm described in Chapter 4, the 3D trajectory of the instrument in each trial was estimated. Figure 5.6 shows the trajectory of the instrument for a single experimental trial. The following two sections describe how the 2D feature data and 3D trajectory data were used to characterize V-PITS.

5.2 Static Noise Characterization

To analyze static noise characteristics of V-PITS, 2D track point data and 3D trajectory data over 90 frames near the end of each trial within a single dataset were analyzed. Over these frames the instrument has already been displaced and is approximately stationary. Therefore variation in the 2D track point position and estimated 3D position are primarily due to noise. The statistical range $(max - min)$ of the 2D track point position and 3D position are used to quantify the noise level. For the 2D track point, the range is calculated for the x and y pixel

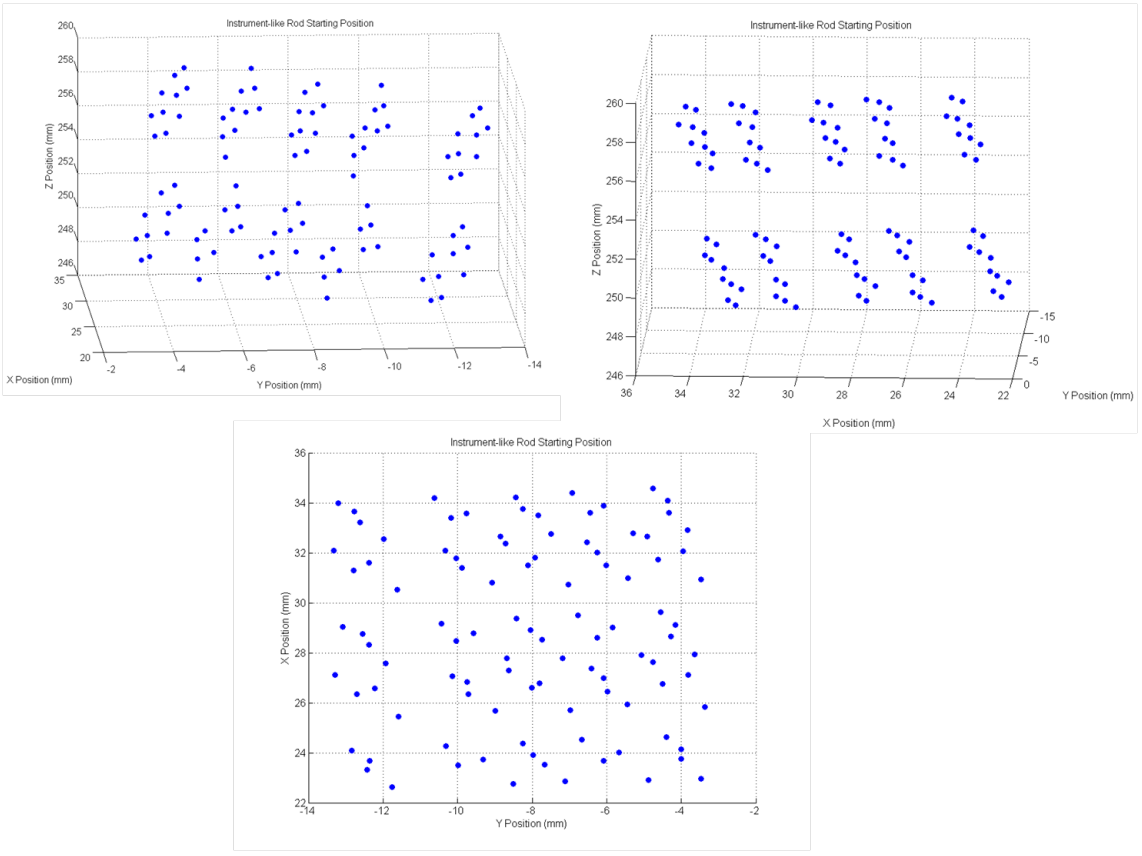

Figure 5.3 Single Dataset Instrument Starting Positions. Each circle represents an individual experimental trial within a dataset. The position of the circle is the starting position of instrument-like rod in the camera frame of the left camera in the stereo camera rig. The three plots are the same set of starting points viewed at different orientations.

Figure 5.4 Instrument Starting Position Mosaic.

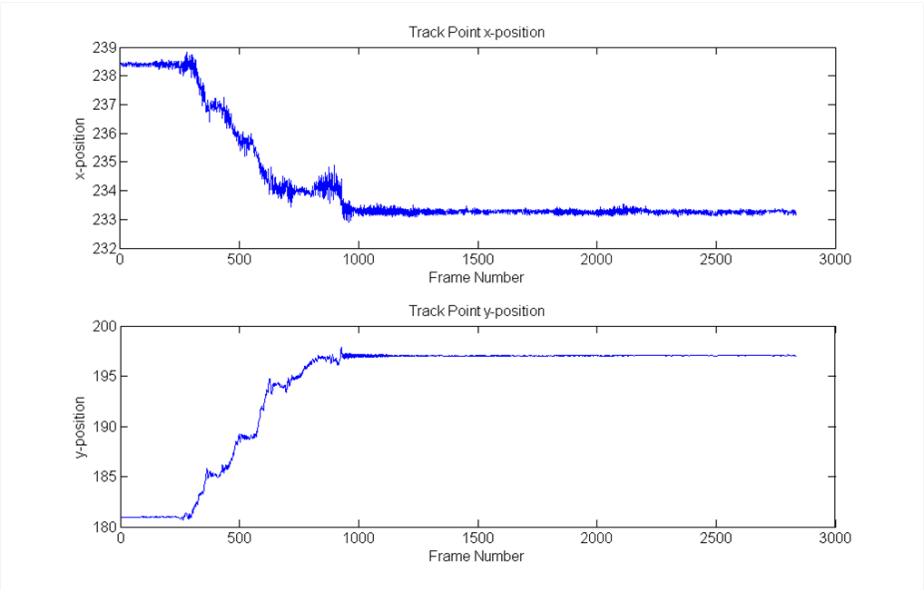

Figure 5.5 Track Point Data for Single Experimental Characterization Trial. The upper plot is the track point x position. The lower plot is the track point y position. Step-like movement is related to manual stage used to displace the instrument.

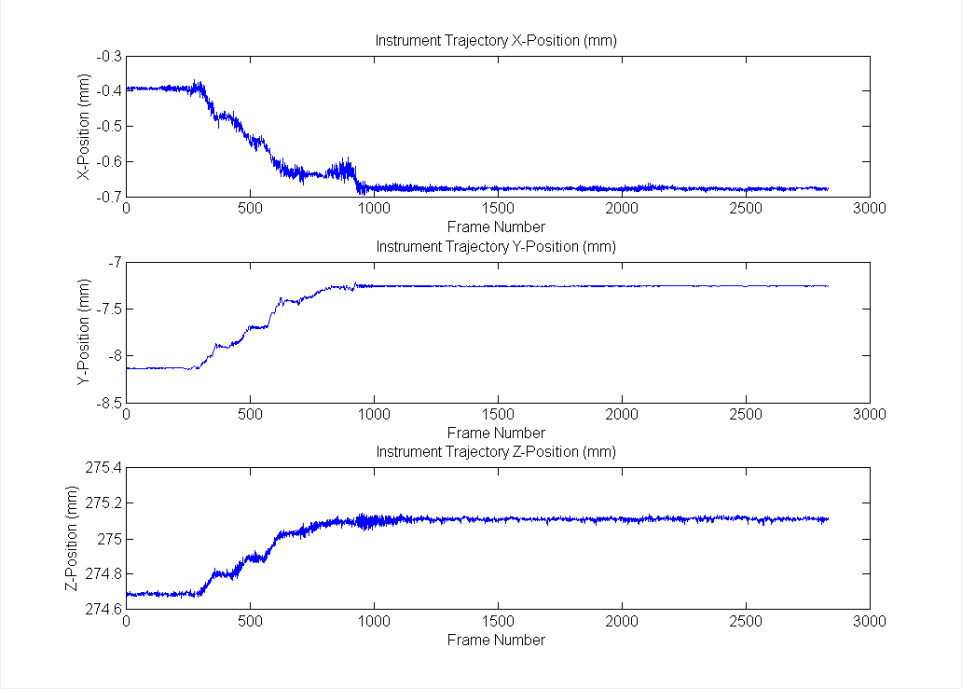

Figure 5.6 Trajectory Data for Single Experimental Characterization Trial. Upper plot is instrument X position in left camera frame. Middle plot is instrument Y position in left camera frame. Lower plot is instrument Z position in left camera frame. Step-like movement is related to the manual stage used to displace the instrument.

	Left		Right	
	x	y	x	y
Mean	0.232	0.0935	0.263	0.123
Median	0.206	0.0887	0.242	0.114
Max	0.725	0.294	0.543	0.315
Min	0.122	0.0473	0.148	0.0701

Table 5.1 2D Static Noise Statistics Across Experimental Trials. All values are in pixels. The column labels x and y refer to the track point position in the x and y image directions. The column labels left and right refer to the left and right camera videos.

position. For the 3D trajectory, the range is calculated for the instrument's $X, Y,$ and Z position (in mm). These axes correspond to the left camera's coordinate frame.

5.2.1 2D Static Noise Characterization

For each trial within a single dataset, the range of track point position values over 90 stationary frames was calculated. Statistics for the range value across all trials (starting positions) within the dataset were calculated and can be seen in Table 5.1. Each statistic was computed for each image direction (x,y) in both camera videos (left,right). The units of each statistic is in pixels. Deviation from zero can be taken as a measure of 2D static noise. In the table, values in the x direction are larger than those in the y direction for all statistics. This indicates that 2D static noise is larger in the x direction. The worst case range (max statistic) is 0.725 pixels in the x direction and 0.315 pixels in the y direction. Therefore, the limiting uncertainty in the 2D track point position based on noise is +/- 0.363 pixels.

Bubble plots were used to evaluate spatial dependence of noise. These plots contain an x and y axis representative of image coordinates. For each trial a bubble is placed at the initial track point position. The size of the bubble is proportional to the magnitude of the range (in a single image direction) for that trial. Figure 5.7 contains bubble plots of range values in the x direction for the left and right view respectively. The bubble plots contain clusters of larger range values for small initial y positions. This corresponds to trials in which the instrument's

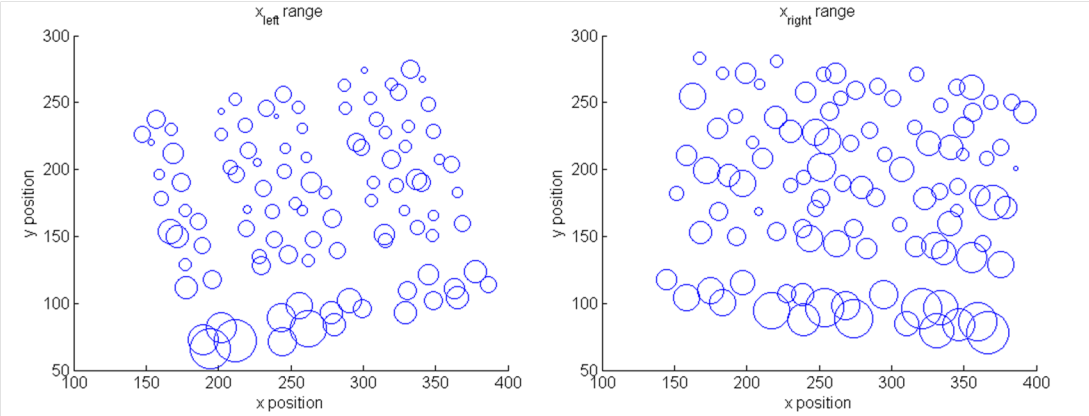

Figure 5.7 Track Point Bubble Plots, x position. The left plot is for the left camera videos. The right plot is for the right camera videos. The size of the bubbles are not to scale of the axes.

marker and track point are near the top of a video frame. An example of this is seen in Figure 5.9. This trend was further examined by plotting the range values in the x direction vs. the initial track point y position as seen in Figure 5.8. This plot is for the left camera videos. An increase in range value in the x direction is seen for decreasing y position.

This increase in noise level is related to instrument midline fitting. In each video frame the tracking algorithm detects the location of a window enclosing the marker attached to the instrument and makes an estimate of the instrument's imaged midline. This midline is estimated by detecting and fitting lines to the visible boundaries above the marker window. The instrument track point position is found by intersecting the midline with a horizontal line at the vertical midsection of the marker window. An overlaid box with cross-hair represents the marker window in Figure 5.9. The lines above the window are the detected boundary lines and estimated imaged midline. Because of the marker's position, only 20 pixels are available to make a boundary line estimate. Due to the limited number of pixels available for fitting, the boundary line parameters are sensitive to noise and quantization. This couples into the midline estimate and the track point position. As the marker window's position is increased in the y direction, the effect of boundary pixel noise is reduced due to more pixels being used for fitting. Figures 5.10 and 5.11 show the left boundary line parameters (θ, ρ) for two different trials.

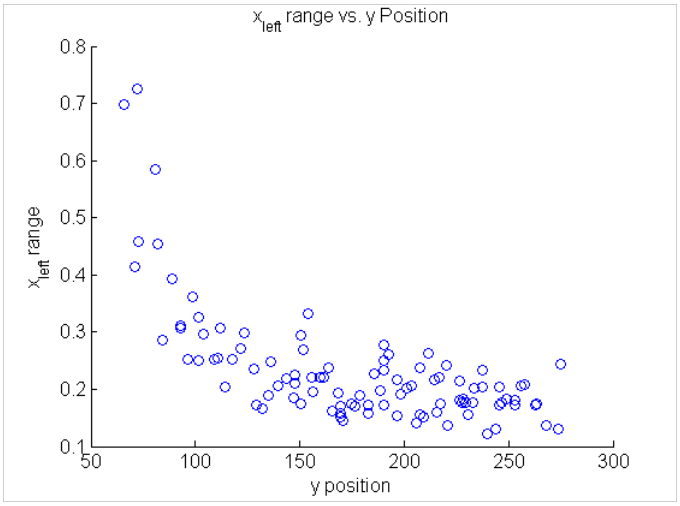

Figure 5.8 Track Point x range vs. Track Point Initial y position. Higher levels of noise (larger range values) are seen for smaller y positions.

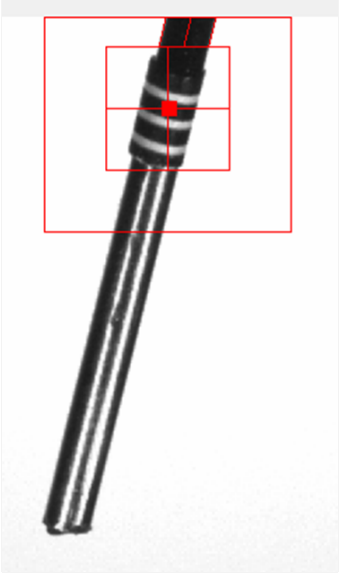

Figure 5.9 Marker Near Top of Image.

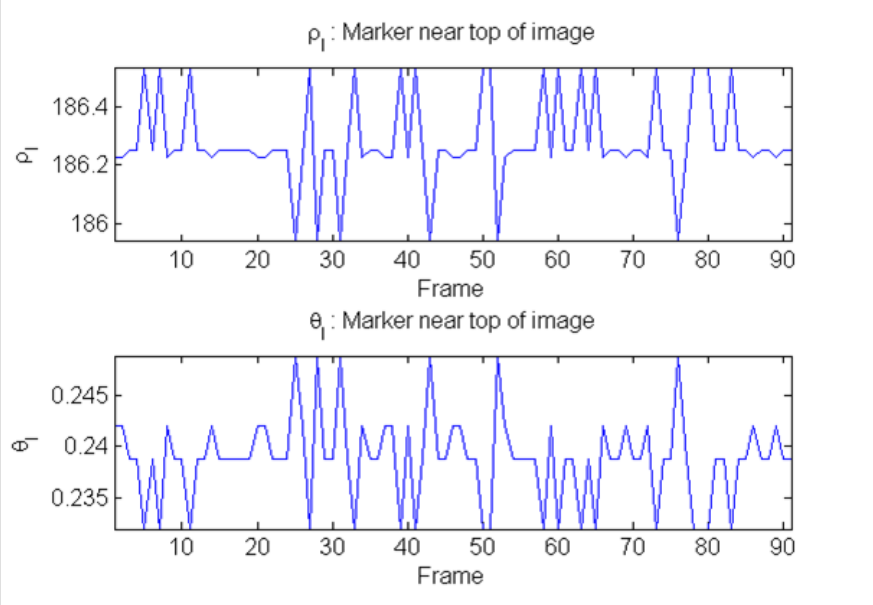

Figure 5.10 Line Parameters, Marker Near Top of Image.

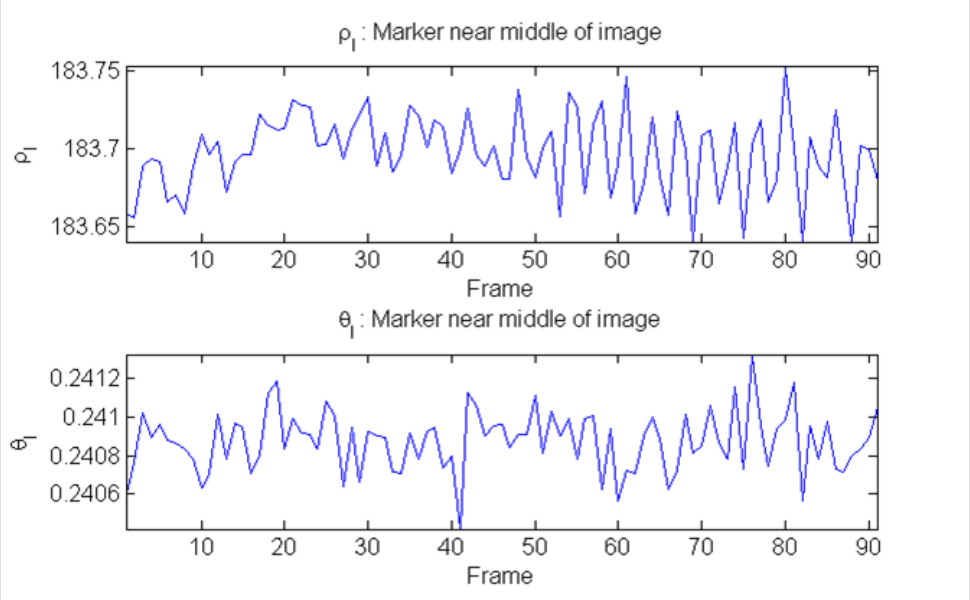

Figure 5.11 Line Parameters, Marker Near Middle of Image.

	Fitting Points	ρ_L range	θ_L range
Marker Near Top of Image	22	0.59	0.0171
Marker Near Middle of Image	177	0.113	0.000908

Table 5.2 Boundary Line Parameter Range Comparison

Figure 5.10 contains the line parameters for a trial in which the track point is near the top of the image. On average, twenty two points are used to fit the left boundary line. Figure 5.11 contains the line parameters for a case in which the track point is near the middle of the image. On average 177 pixels are used to fit the boundary line. A comparison of the range taken by the line parameters is given in Table 5.2. The line parameters are more sensitive to boundary pixel noise when fewer points are used for fitting. This can be seen by comparing the values in the table. When 177 boundary pixels are used the ρ range is 0.113. For 22 pixels it is more than five times this at 0.59. For θ, a range one order of magnitude separates the line estimated with 22 pixels and 177 pixels. It is important to note that the boundary lines were fit using a set of pixel locations. By estimating boundary points at a subpixel resolution, noise levels could potentially be reduced.

5.2.2 3D Static Noise Characterization

For each trial within a single dataset, the instrument-like rod's 3D trajectory was estimated using the algorithm described in Chapter 4. 3D static noise was characterized by examining 90 frames near the end of each trial. Over these frames the instrument is approximately stationary. For each trial, the range of each 3D position coordinate was calculated. Statistics for the range values across all trials (starting positions) are given in Table 5.3. The coordinate axis labels X, Y, Z refer to the coordinate frame of the left camera in the stereo camera rig. The statistic values in the Z direction are larger than those in the X and Y direction. The maximum Z range is 0.145 mm compared to 0.0268 mm for the X direction and 0.0160 mm for the Y direction. The uncertainty due to static noise for the $X, Y,$ and Z directions is +/- 0.0134 mm, +/- 0.0080 mm, and +/- 0.0725 mm.

	X	Y	Z
Mean	0.0127	0.00565	0.0420
Median	0.0119	0.0055	0.0350
Max	0.0268	0.0160	0.145
Min	0.00729	0.00330	0.00180

Table 5.3 3D Static Noise Statistics Across Trials. Column labels X, Y, Z refer to the left camera's coordinate frame. All values are in mm.

The 2D static noise described in the previous subsection couples into the 3D trajectory estimate. The values in the table show that the 3D position in the Z direction is more sensitive to 2D static noise than the X and Y direction. A histogram of the Z range values for all trials is seen in Figure 5.12. The lowest bin contains the majority of the range values. Similarly to the previous section, when few pixels are available for boundary line estimates noise level (range value) increases. These trials correspond to the higher non-zero bins of the histogram.

5.3 Accuracy Evaluation

Three datasets of experimental trials were collected. In each trial, video was captured of the instrument-like rod being displaced 1 mm. The direction of displacement was fixed within each dataset. To characterize displacement measurement accuracy, instrument displacement was calculated using the estimated trajectory. Comparing the actual displacement (1 mm) to this calculated displacement allows the accuracy to be characterized. Let $\mathbf{T}(n) = \begin{bmatrix} X_{traj}(n) & Y_{traj}(n) & Z_{traj}(n) \end{bmatrix}^T$ be the trajectory of the instrument-like rod where n is the frame number. Assume this trajectory was found using the trajectory estimation algorithm and is in the coordinate frame of the left camera. The displacement signal is given by

$$D(n) = \sqrt{(X_{traj}(n) - X_{traj}(1))^2 + (Y_{traj}(n) - Y_{traj}(1))^2 + (Z_{traj}(n) - Z_{traj}(1))^2}. \quad (5.1)$$

The displacement error signal is given by

$$D_{error}(n) = 1 - D(n). \quad (5.2)$$

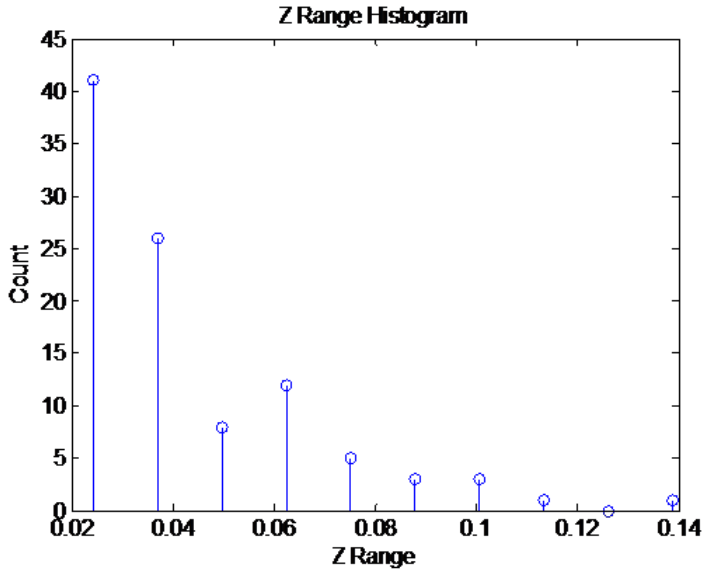

Figure 5.12 Instrument Trajectory, Z Range Histogram.

The value of this signal is only applicable for frames after the displacement event has been completed. Assume the instrument has been completely displaced and vibrations due to movement have dampened out prior to the frame subset n_{start} to n_{end}. The RMS displacement error ($RMSE$) is given by

$$RMSE = \sqrt{\frac{1}{1 + (n_{end} - n_{start})} \sum_{n=n_{start}}^{n_{end}} D_{error}(n)^2}. \tag{5.3}$$

The $RMSE$ was calculated for each trial using 90 frames near the end of the trial. Table 5.4 contains $RMSE$ value statistics computed across all trials (starting positions) within a single dataset. The three right columns correspond to the different datasets collected. The displacement direction of the instrument-like rod in a dataset is given by the label in the first row. The axis labels X,Y, and Z correspond to the directions in Figure 5.2. In the table, the Y displacement always has the largest values. The worst case $RMSE$ for a dataset is given by the max statistic. The Y displacement worst case $RMSE$ is 0.1549 mm RMS. This is 1.86x the worst case Z displacement $RMSE$ and 4.20x the worst case X displacement $RMSE$.

A histogram of the individual Y displacement $RMSE$ values is seen in Figure 5.13. The majority of values fall within bins below 0.1 mm RMS. It was seen in the previous section that

	X	Y	Z
Mean	0.0161	0.0570	0.0371
Median	0.0145	0.0554	0.0348
Max	0.0369	0.1549	0.0835
Min	0.0017	0.0130	0.0114

Table 5.4 RMSE Statistics Across Trials. Column labels X,Y,Z refer to the instrument displacement direction. All values are in mm RMS.

decreases in the marker position resulted in increased static noise levels. Figure 5.14 is a scatter plot of an individual trial's $RMSE$ with respect to the initial track point y position in the left camera videos. It is seen in this plot that the worst case $RMSE$ occured at a trial with small y position. But, this error appears as an outlier as it has a value of 0.1549 mm RMS and all other trials are below 0.1 mm RMS. While static noise likely had influence on this datapoint, it is unlikely that it is the primary reason for the large $RMSE$.

5.4 Chapter Conclusion

Two dimensional static track point noise was found to be below +/- 1 pixel. The limiting uncertainty of the 2D track point position due to static noise was found to be +/- 0.363 pixels. Marker position with respect to the top of the video frame affects this noise level. As the marker moves further from the top of the frame, more pixels are available to fit the instrument's boundary lines and the effect of static noise is reduced. This 2D noise couples into the estimated 3D instrument trajectory. For 3D position, a limiting uncertainty due to static noise of +/- 0.0725 mm was found. This limiting uncertainty was measured in the instrument's Z coordinate in the left camera frame. Worst case uncertainties were smaller for the X and Y direction. Therefore static noise has the most influence on the position of an instrument along the depth of a camera. The worst case 3D displacement accuracy of the system was found to be 0.1549 mm RMS.

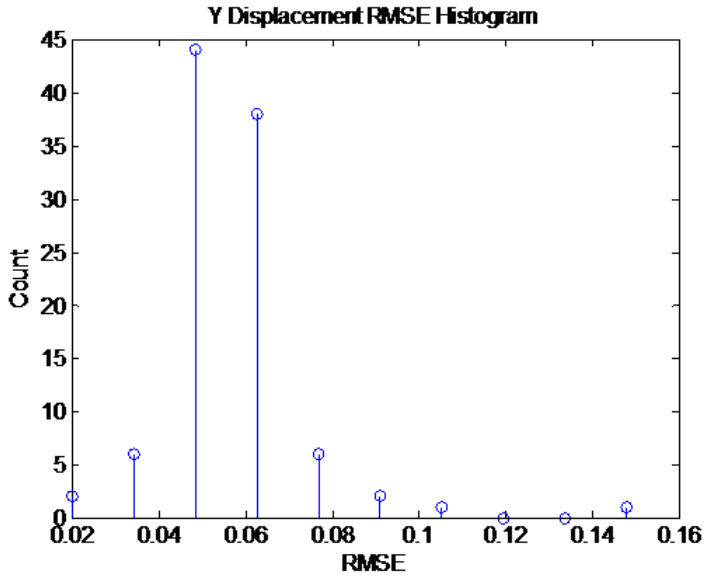

Figure 5.13 Y Displacement Dataset RMSE Histogram.

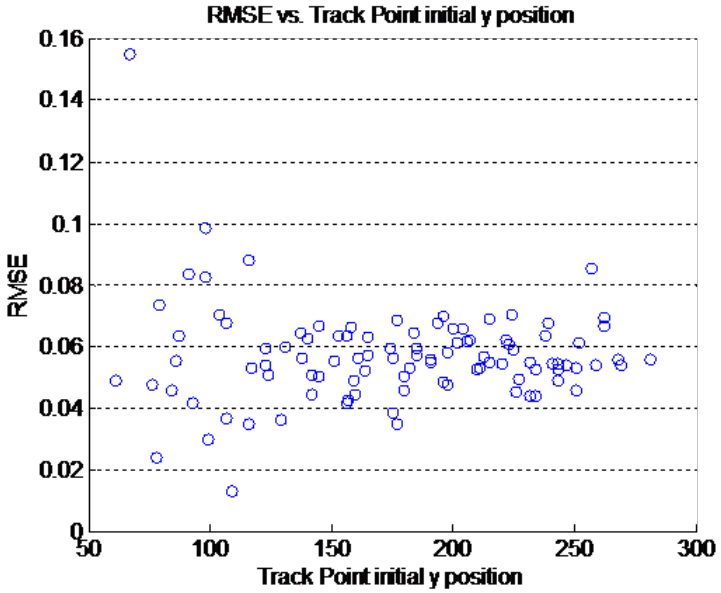

Figure 5.14 Y Displacement Dataset RMSE vs. Track Point initial y position. The worst case trial appears as an outlier as all other trials are bounded by a value of 0.1 mm RMS

Chapter 6

Conclusion

V-PITS (Video Based Phonomicrosurgery Instrument Tracking System) was developed to estimate the trajectory of instruments performing a simulated phonomicrosurgery exercise. It first captures a synchronized pair of videos using a calibrated stereo camera rig. Next, it processes both videos with a tracking algorithm that detects a set of features in each frame related to instrument position and orientation. Finally, a trajectory estimation algorithm uses the features in both videos to estimate the 3D trajectory. Uncertainty in the position reported by the system due to noise was characterized as +/- 0.073 mm. Displacement measurement accuracy was characterized as 0.15 mm RMS.

Video acquisition software was written using C++. An application with a GUI was written that allowed for synchronized capture of video from the stereo camera rig. The camera manufacturer's SDK (Basler Pylon) was used to communicate with and grab frame data from the cameras. The toolkit wxWidgets (http://www.wxwidgets.org) was used to implement the GUI. All algorithmic components were developed using Matlab. The instrument tracking algorithm was implemented in a semi-automated fahsion as described in Subsection 3.3.3 of Chapter 3. A GUI application was written for this. A function was written that calculated 3D position data using the trajectory estimation algorithm.

6.1 Future Work and Optimization

This project was motivated by a past project that involved the quantification of technical surgical skill. In the future, V-PITS will be used for experiments related to technical phonomicrosurgery skill. The instrument motion data reported by V-PITS will be be further processed to quantify movement efficiency and control. Possible future studies include: a comparison of surgeons at different levels of experience, quantification of learning curve, and evaluation of external factors like arm-support type and microscope type.

Algorithm speed is primarily limited by the template matching scheme used during instrument tracking. This could be sped up by first finding the template position at a lower resolution. Alternatively, an initial estimate of the template position could be found using binary template matching. Speed is an issue because tracking is performed in a semi-automated fashion. Therefore a user must wait as the tracking algorithm runs. Speeding up the algorithm would reduce this waiting time. This waiting time could also be eliminated if the algorithm was completely automated. As the project progressed, the quality of the videos captured also progressed. This was primarily due to improvements in the lighting methodology used. This has reduced the variability in instrument appearance in frames that the algorithm can correctly detect features. This makes full automation more likely. In such case, it may be necessary to simultaneously track both surgical instruments. Another issue with the system is the amount of hard-drive memory used to store videos. Currently videos are saved using the lossless video codec Huffyuv (http://neuron2.net/www.math.berkeley.edu/benrg/huffyuv.html). The influence of a lossy video codec (with higher compression) on system accuracy and noise would have to be investigated prior to be being used.

LIST OF REFERENCES

[1] Jean-Yves Bouguet. Camera calibration toolbox for matlab, July 2010.

[2] Olivier Faugeras. *Three-dimensional computer vision: a geometric viewpoint*. MIT Press, Cambridge, MA, USA, 1993.

[3] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.

[4] W.J. Gould, R.T. Sataloff, and J.R. Spiegel. *Voice surgery*. Mosby, 1993.

[5] Teodor P. Grantcharov, Linda Bardram, Peter Funch-Jensen, and Jacob Rosenberg. Assessment of technical surgical skills. *European Journal of Surgery*, 168(3):139–144, 2002.

[6] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, New York, NY, USA, 2000.

[7] J. P. Lewis. Fast normalized cross-correlation, 1995.

[8] Yi Ma, Stefano Soatto, Jana Kosecka, and S. Shankar Sastry. *An Invitation to 3-D Vision: From Images to Geometric Models*. SpringerVerlag, 2003.

[9] Mark Nixon and Alberto S. Aguado. *Feature Extraction & Image Processing, Second Edition*. Academic Press, 2nd edition, 2008.

[10] Carol Reiley, Henry Lin, David Yuh, and Gregory Hager. Review of methods for objective surgical skill evaluation. *Surgical Endoscopy*, 25:356–366, 2011. 10.1007/s00464-010-1190-z.

[11] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition, 2010.

[12] Steven M. Zeitels and Gerald B. Healy. Laryngology and phonosurgery. *New England Journal of Medicine*, 349(9):882–892, 2003.