

---


Electronic Theses and Dissertations, 2004-2019

---

2004

## Conformal Tracking For Virtual Environments

Larry Dennis Davis Jr.  
*University of Central Florida*

 Part of the [Electrical and Computer Engineering Commons](#)  
Find similar works at: <https://stars.library.ucf.edu/etd>  
University of Central Florida Libraries <http://library.ucf.edu>

This Doctoral Dissertation (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Electronic Theses and Dissertations, 2004-2019 by an authorized administrator of STARS. For more information, please contact [STARS@ucf.edu](mailto:STARS@ucf.edu).

---

### STARS Citation

Davis, Larry Dennis Jr., "Conformal Tracking For Virtual Environments" (2004). *Electronic Theses and Dissertations, 2004-2019*. 85.  
<https://stars.library.ucf.edu/etd/85>

# CONFORMAL TRACKING FOR VIRTUAL ENVIRONMENTS

by

LARRY D. DAVIS, JR.

B.S.E.E. Florida Agricultural and Mechanical University, 1995

M.S.E.E. University of Central Florida, 1999

A dissertation submitted in partial fulfillment of the requirements  
for the degree of Doctor of Philosophy  
in the Department of Electrical & Computer Engineering  
in the College of Engineering & Computer Science  
at the University of Central Florida  
Orlando, Florida

Spring Term  
2004

Major Professor: Jannick P. Rolland

© 2004 Larry D. Davis, Jr.

## ABSTRACT

A virtual environment is a set of surroundings that appears to exist to a user through sensory stimuli provided by a computer. By virtual environment, we mean to include environments supporting the full range from VR to pure reality. A necessity for virtual environments is knowledge of the location of objects in the environment. This is referred to as the tracking problem, which points to the need for accurate and precise tracking in virtual environments.

Marker-based tracking is a technique which employs fiduciary marks to determine the pose of a tracked object. A collection of markers arranged in a rigid configuration is called a tracking probe. The performance of marker-based tracking systems depends upon the fidelity of the pose estimates provided by tracking probes.

The realization that tracking performance is linked to probe performance necessitates investigation into the design of tracking probes for proponents of marker-based tracking. The challenges involved with probe design include prediction of the accuracy and precision of a tracking probe, the creation of arbitrarily-shaped tracking probes, and the assessment of the newly created probes.

To address these issues, we present a pioneer framework for designing conformal tracking probes. Conformal in this work means to adapt to the shape of the tracked objects and to the environmental constraints. As part of the framework, the accuracy in position and orientation of a given probe may be predicted given the system noise. The framework is a methodology for designing tracking probes based upon performance goals and environmental constraints.

After presenting the conformal tracking framework, the elements used for completing the steps of the framework are discussed. We start with the application of optimization methods for determining the probe geometry. Two overall methods for mapping markers on tracking probes are presented, the Intermediary Algorithm and the Viewpoints Algorithm.

Next, we examine the method used for pose estimation and present a mathematical model of error propagation used for predicting probe performance in pose estimation. The model uses a first-order error propagation, perturbing the simulated marker locations with Gaussian noise. The

marker locations with error are then traced through the pose estimation process and the effects of the noise are analyzed. Moreover, the effects of changing the probe size or the number of markers are discussed.

Finally, the conformal tracking framework is validated experimentally. The assessment methods are divided into simulation and post-fabrication methods. Under simulation, we discuss testing of the performance of each probe design. Then, post-fabrication assessment is performed, including accuracy measurements in orientation and position. The framework is validated with four tracking probes. The first probe is a six-marker planar probe. The predicted accuracy of the probe was 0.06 deg and the measured accuracy was  $0.083 \pm 0.015$  deg. The second probe was a pair of concentric, planar tracking probes mounted together. The smaller probe had a predicted accuracy of 0.206 deg and a measured accuracy of  $0.282 \pm 0.03$  deg. The larger probe had a predicted accuracy of 0.039 deg and a measured accuracy of  $0.017 \pm 0.02$  deg. The third tracking probe was a semi-spherical head tracking probe. The predicted accuracy in orientation and position was  $0.54 \pm 0.24$  deg and  $0.24 \pm 0.1$  mm, respectively. The experimental accuracy in orientation and position was  $0.60 \pm 0.03$  deg and  $0.225 \pm 0.05$  mm, respectively. The last probe was an integrated, head-mounted display probe, created using the conformal design process. The predicted accuracy of this probe was  $0.032 \pm 0.02$  degrees in orientation and  $0.14 \pm 0.08$  mm in position. The measured accuracy of the probe was  $0.028 \pm 0.01$  degrees in orientation and  $0.11 \pm 0.01$  mm in position. These results constitute an order of magnitude improvement over current marker-based tracking probes in orientation, indicating the benefits of a conformal tracking approach. Also, this result translates to a predicted positional overlay error of a virtual object presented at 1m of less than 0.5 mm, which is well above reported overlay performance in virtual environments.

*To The Ancestors: For allowing me to be your hope and dream,*  
*To My Mother: For life and teaching me to be the person I am,*  
*To My Father: For inspiration and providing the template for manhood,*  
*To My Brothers: For teaching me patience and “Helping the Bear,”*  
*To Bernadette: For love and support – mental and financial,*  
*To Elden: For constantly reminding me what’s really important in life, and*  
*To the Davises To Come: For being the twinkle in our eyes...*

## **In Memorial**

*To Edwin Senechal: For your spirit,*

*and*

*To Dr. Israel Tribble, Jr.:*

*For teaching us “If you can walk, you can dance; If you can talk, you can sing.”*

*-I am the master of my fate; I am the captain of my soul.-*

## ACKNOWLEDGMENTS

There is a Ugandan proverb that says “A child does not grow up only in a single home.” In the sense of my finishing the Ph.D. degree, I am truly a child of many homes, whose residents encouraged, taught, and sustained me along the way. If I were to list everyone who has contributed to this degree, I would need to add another chapter to this work. However, protocol binds me to make my writing brief, and I offer a heartfelt “thank you” to those whom I am unable to name here.

I start by thanking my advisor, Jannick Rolland, for the opportunity to pursue a Ph.D. under her guidance. Under her tutelage, I have learned the importance of thoughtful design, assessment, and “making the story sexy.” *Merci bien pour votre enseignement.* Next, I thank my committee, as each member has inspired me in addition to guiding the writing of this dissertation. My thanks to:

- Charles Hughes for being a hacker at heart, despite the demands of professorship.
- Wasfy Mikhael for teaching the best DSP courses and having a world view that we would all do well to imitate.
- J. Michael Moshell for writing the summary of VR projects at IST that brought me here and being a visionary thinker.
- Eric Clarkson for suffering my mathematical ineptitude and being the first contractor as part of the Mathematics Consultants Group.

I also thank Felix Hamza-Lup for working closely with me on the mapping algorithms and other members of the scientific community who have contributed to or influenced this work: Andrei State, Greg Welch, Dannette Allen, Bernard Peuchot, and Boris Zeldovich among others.

Finally, I have been blessed with an inordinate amount of friends, colleagues, and mentors who have helped me “shoulder the load.” I thank:

- The Griffith family for lessons on life and championship Scrabble.

- Mohamed Elhassan for always reminding me about Pin 13.
- Tiffani Williams for your sense of humor and the advice about that class I ignored.
- Keith Hunter for being one of the most well rounded, cerebral people I know.
- Karen Oliver for your support and suffering through the Qualifier Exam.
- Charmaine Harris for being a great office-mate and singing 102 Jamz songs.
- Dr. Jacqueline Smith for reminding me of the work to be done in the larger scope.
- The Mighty McKnight Fellows: Damon Bryant, Erica Wells, T. Nichole Phillips, and Kenyatta Rivers for advice, a place to gripe, and lunch.
- The ECL'ers: Marc Smith, Paulius Micikevicius, and Lynda Vidot for awesome discussions and adding a year to my tenure at UCF.
- The Most Esteemed Colleagues of the Epic Gathering: Andrey Krywonos, Javier Gonzales, Gabriel "CB" Popescu, Vesselin Chaoulov, and Eric Nelson for camaraderie, racquetball, a Wilderness of Frogs, and 80's music at Gators.
- Office 171: Cali Fidopiastis and Felix Hamza-Lup for their support.
- The Florida Education Fund: Dr. Lawrence Morehouse, Charles Jackson, the Administrative Staff, and Dr. Dovie Gamble
- The ODALab, Past and Present for assistance at various stages in my program.

The work presented in this dissertation was funded, in part, by the National Institute of Health, the National Science Foundation, the US Army STRICOM, and the Florida Education Fund.



# TABLE OF CONTENTS

LIST OF FIGURES	xi
LIST OF TABLES	xiii
CHAPTER 1: INTRODUCTION	1
1.1 Tracking Technology in Virtual Environments . . . . .	3
1.2 Motivation . . . . .	6
1.3 Research Summary . . . . .	7
1.4 Dissertation Outline . . . . .	7
CHAPTER 2: RELATED WORK	9
2.1 Marker-Based Tracking Algorithms . . . . .	9
2.2 Pose Estimation Algorithms . . . . .	12
2.3 Previous Research in Pose Error Determination . . . . .	13
CHAPTER 3: CONFORMAL PROBE DESIGN	16
3.1 A Procedure for Conformal Probe Design . . . . .	17
3.2 Environmental and Application Factors . . . . .	19
3.3 Marker Placement Methods . . . . .	23
3.3.1 The Modified Simulated Annealing Algorithm . . . . .	24
3.3.2 Conformal Mapping for Arbitrary Objects . . . . .	28
3.4 Metrics for Probe Performance . . . . .	35
CHAPTER 4: POSE ESTIMATION	40
4.1 Determining a Coordinate Frame . . . . .	40
4.1.1 Geometric Frame Determination . . . . .	41
4.1.2 PCA-based Frame Determination . . . . .	42

4.1.3	Extension to Probes with Hidden Markers . . . . .	43
4.2	Pose Estimation . . . . .	45
4.2.1	Direct Computation of Pose . . . . .	45
4.2.2	Optimization Approaches: Least Squares Estimation . . . . .	47
4.3	Error Analysis in Pose Estimation . . . . .	51
CHAPTER 5: THE EFFECTS OF ENVIRONMENTAL AND APPLICATION FACTORS		
	ON POSE ESTIMATION	53
5.1	Noise Propagation in Pose Estimation . . . . .	53
5.2	Effects of Factors on Pose Estimation . . . . .	65
5.2.1	The Effect of Tracker Accuracy and Frame Determination on Pose Estimation	67
5.2.2	The Effect of Probe Size and Shape on Pose Estimation . . . . .	73
5.2.3	The Effect of the Number of Markers and Marker Field of Emission on Pose Estimation . . . . .	78
5.2.4	Matrix of Tracking Factors . . . . .	79
CHAPTER 6: APPLICATION TO TRACKING PROBES		81
6.1	Simulation of Tracking Probes . . . . .	81
6.1.1	Simulation Specifics . . . . .	84
6.2	Experimental Results from Tracking Probes . . . . .	86
6.2.1	A Six-Marker Planar Probe . . . . .	88
6.2.2	Two Concentric Tracking Probes . . . . .	89
6.3	Case Study: Design, Implementation, and Assessment of Conformal Probes . . . . .	96
6.3.1	A Semi-spherical Head Tracking Probe . . . . .	96
6.3.2	A Conformally Mapped Probe – HMD . . . . .	99
CHAPTER 7: DISCUSSION AND CONCLUSIONS		103
7.1	Further Insights for Conformal Probe Design . . . . .	103

7.2	Marker Mapping Concerns . . . . .	105
7.3	Numerical Limitations . . . . .	105
7.4	Marker/Tracker Issues . . . . .	107
7.5	Scale Factor in Pose Estimation . . . . .	107
7.6	Future Work . . . . .	107
APPENDIX SOLUTION OF EIGENVALUE COMPONENTS		111
LIST OF REFERENCES		114

## LIST OF FIGURES

1.1	From Real to Virtual: The Virtuality Continuum . . . . .	1
3.1	An Algorithm for Conformal Probe Design . . . . .	20
3.2	Iterative Steps of Conformal Probe Design . . . . .	21
3.3	The Modified Simulated Annealing Algorithm . . . . .	27
3.4	The Intermediary Algorithm . . . . .	31
3.5	The Intersection Determination Algorithm . . . . .	33
3.6	The Viewpoint Algorithm . . . . .	36
3.7	Detail of the UpdateTriangles Routine . . . . .	37
5.1	Semilog plot of the Norm of the A Matrix versus the Amount of Tracker Noise . . . . .	72
5.2	Semilog plot of the Norm of the A Matrix versus the Size of the Probe . . . . .	76
6.1	MATLAB Simulation GUI . . . . .	85
6.2	Data Points Fit with a Gaussian . . . . .	87
6.3	A Six-marker Planar Probe . . . . .	88
6.4	Simulation Results from a Six-Marker Planar Tracking Probe . . . . .	89
6.5	Simulation Results from a Six-Marker Planar Tracking Probe (at close range) . . . . .	90
6.6	Two Concentric Tracking Probes . . . . .	91
6.7	Simulation Results from Small Concentric Tracking Probe . . . . .	92
6.8	Simulation Results from Small Concentric Tracking Probe (at close range) . . . . .	93
6.9	Simulation Results from Large Concentric Tracking Probe . . . . .	94
6.10	Simulation Results from Large Concentric Tracking Probe (at close range) . . . . .	95
6.11	A Semi-spherical Head Tracking Probe . . . . .	96
6.12	Simulation Results from Semi-spherical Head Tracking probe . . . . .	97
6.13	Simulation Results from Semi-spherical Head Tracking Probe (at close range) . . . . .	98

6.14 VRML Visualization of a Conformally Mapped Tracking Probe . . . . .	99
6.15 Realization of a Conformally Mapped Tracking Probe . . . . .	100
6.16 Simulation Results of the HMD Tracking Probe . . . . .	101
6.17 Simulation Results of the HMD Tracking Probe (at close range) . . . . .	102
7.1 A Plot of Pose Error vs. large numbers of markers . . . . .	104

## LIST OF TABLES

5.1	Matrix of Tracking Factors . . . . .	79
6.1	Marker Placements for a Spiral, Planar Tracking Probe . . . . .	86

## CHAPTER 1: INTRODUCTION

A virtual environment is a set of surroundings that appears to exist to a user through sensory stimuli provided by a computer. Virtual environments may be classified along a virtuality continuum shown in Figure 1.1, that ranges from the real environment to a completely computer-generated environment [Milgram and Kishino, 1994].

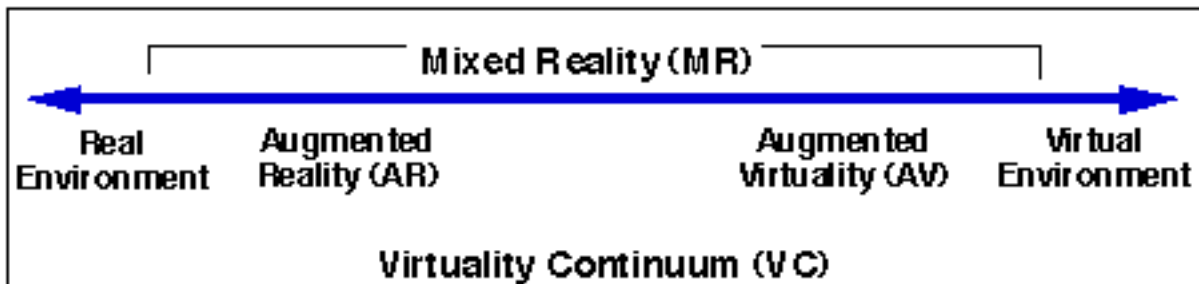


Figure 1.1: From Real to Virtual: The Virtuality Continuum

The continuum proposed by Milgram and Kishino, purely in the context of visual displays, can naturally be extended to multi-modal virtual environments, that is, environments with combinations of different types of stimuli. An application can therefore be classified within the continuum according to its amount of computer-generated stimuli.

Virtual reality (VR) environments are generated entirely by computer and require immersive displays, which only provide a synthetic view of the world. Mixed reality (MR) environments combine varying amounts of computer-generated stimuli with the real environment. MR includes augmented virtuality (AV), where computer-generated stimuli predominate, and augmented reality (AR), where the majority of stimuli is present from the real environment. MR environments require see-through displays due to the added condition of presenting virtual stimuli in spatial coincidence with real objects, known as registration [Rolland and Fuchs, 2000].

The terms “Virtual Reality”, created by Jaron Lanier, and “Artificial Reality”, coined by Myron Krueger, were the first widely used terms applied to computer-generated synthetic environments

[Rheingold, 1991]. Although the terms came into use in the late 70's and early 80's, the first virtual environments were created almost a decade earlier.

The first attempt at creating a virtual environment was the Sensorama by Morton Heilig in 1960 [Rheingold, 1991]. The Sensorama relied entirely upon film sequences instead of computers and incorporated vibrations and scents to provide a feeling of immersion in various situations. However, the first virtual environment, an augmented reality navigational aid for helicopter pilots, was created by Ivan Sutherland [Sutherland, 1968]. Sutherland's initial system used CRT displays attached to a mechanical arm (which also served to track the user), later evolving into a unlinked system with ultrasonic tracking. Myron Krueger developed a video-based virtual environment that used back-projection techniques to place users within the environment [Krueger, 1977]. Krueger later developed the VIDEOPLACE system, which used computer vision techniques to track users within the environment to generate visual and auditory responses [Krueger, 1985]. Jaron Lanier and Thomas Zimmerman founded the VR industry with the creation of VPL Inc. in 1984 [Rheingold, 1991]. VPL was formed at a crucial time for virtual environments; VR was a hot technology, computer power was exponentially increasing, personal computing was becoming ubiquitous, and increased defense spending pushed more money into the research sector. As a result, VPL products ended up in most of the virtual environment research labs in the world. VPL was also responsible for creating the DataGlove [Zimmerman and Lanier, 1987]. In close partnership with VPL, Scott Fisher et al. also developed a computer-generated, multi-sensory stereoscopic virtual environment that incorporated gestural input [Fisher et al., 1986]. Gestural input was obtained from a DataGlove and the environment also incorporated sound. As a follow up to the ideas introduced by Krueger, Carolina Cruz-Neira et al. created a projection-based Cave Automated Virtual Environment (CAVE) [Cruz-Neira et al., 1993]. The CAVE immersed users in a 3D environment using shutter glasses and video projection techniques. An extension of this technology, the Immersadesk, was presented in [Czernuszenko et al., 1997].



There were also pioneering efforts taking place in mixed reality. As mentioned previously, Sutherland's groundbreaking virtual environment was an AR system. Andrew Lippman also developed a video-based, MR environment that used pre-recorded video sequences to provide a virtual ride through Aspen, Colorado [Lippman, 1980]. The system accessed the video sequences from video disk and responded to user input through a touch screen. The first medical AR system was conceived by Henry Fuchs in the late 1980's and developed by Bajura et al. in 1992 [Bajura et al., 1992]. This research effort aimed at providing *in vivo* fetal visualization via ultrasound. While *in vivo* 3D superimposition was an ultimate goal of the application, Bajura et al. demonstrated the feasibility of superimposing virtual images within a clinical setting. Tom Caudell and David Mizell developed a system to investigate how AR may aid in aircraft assembly [Caudell and Mizell, 1992]. The system was developed at Boeing and superimposed wiring on the field of view of the user. Furthermore, Steven Feiner et al. developed an AR system to provide insight into printer maintenance [Feiner et al., 1993].

### 1.1 Tracking Technology in Virtual Environments

Virtual environments require interaction. The minimum requirement for interaction is knowledge of the location of the user in the environment. Furthermore, if there is to be interaction with other objects in the environment, then their locations must be known as well. Thus, real objects must be tracked within a virtual environment. In addition, the tracking methodology used within a virtual environment must insure accurate and precise tracking to present objects from the correct viewpoint and to accurately register real and virtual objects.

Virtual environment tracking systems have been classified in a number of surveys [Ferrin, 1991] [Meyer et al., 1992] [Burdea and Coiffet, 1994] [Durlach and Mavor, 1994] [Rolland et al., 2000] [Welch and Foxlin, 2002]. According to the technological principle upon which they operate, the categories of tracking systems are time-frequency measurement, spatial scan, inertial sensing, mechanical linkages, direct-field sensing, and hybrids thereof [Rolland et al., 2000].

Time-frequency measurement trackers measure the time of propagation of a signal, compare the phase difference of a measured signal to a reference, or use frequency measurement techniques to indirectly measure time differences. By taking advantage of a priori knowledge of the system configuration, these systems can be used to extract relative or absolute position and orientation data. Examples of time-frequency systems are ultrasonic trackers [Sutherland, 1968] [Logitech Inc., 1992] and the Global Positioning System, invented by Ivan Getting [Farrell and Barth, 1999].

Spatial scan tracking systems employ optical devices to determine the position and orientation of an object by scanning a working volume. Spatial scan trackers compute the position and the orientation of a target by either analyzing 2D projections of image features or by the determination of sweep-beam angles. A sub-classification of spatial scan systems is outside-in versus inside-out. Wang et al. first proposed this terminology for a subclass of optical trackers that use beacons as target features [Wang et al., 1990]. The subclass was subsequently extended to include pattern recognition and beam sweeping systems to indicate and emphasize their common physical principles [Rolland et al., 2000]. In the outside-in configuration, the sensors are fastened to a fixed reference. In the inside-out configuration, the sensors are attached to a mobile target. A few examples of the many spatial scan tracking systems are the Northern Digital Inc. OPTOTRAK 3020 [Northern Digital Inc., 1999], the 3rdTech Inc. Hi-Ball Tracker [3rdTech Inc., 2002], and the Ascension laserBIRD [Ascension Corp., 2002].

Inertial tracking systems use sensors to measure acceleration and rotation relative to the inertial reference frame of the Earth. The sensor data is then used to determine the absolute position and orientation of an object. Inertial sensors are not typically used as a sole source for object tracking; they are generally incorporated within hybrid tracking systems. However, two examples of inertial tracking systems are the InertiaCube by Intersense, Inc. [Foxlin and Durlach, 1994] and GyroTrac by VR Systems.

As the name implies, mechanical linkage tracking systems uses physical links between the reference and the target. Two types of linkages are used in mechanical tracking systems. The first type is an assembly of mechanical parts that provides the user with multiple rotation capabilities and the orientation of the linkages is computed from the various linkages angles measured with incremental encoders or potentiometers. The other type of mechanical linkage uses wires that are rolled in coils. A spring system ensures tension is applied to the wires to measure the distance accurately. The degrees of freedom of mechanical linkage trackers depend upon the mechanical structure of the tracker. While six degrees of freedom are most often provided, typically only a limited range of motions is possible because of the kinematics of the joints and the length of each link. Mechanical linkage trackers have been successfully integrated into force-feedback systems used to make the virtual experience more interactive [Sutherland, 1968] [Brooks et al., 1990]. In addition, fiber optic sensors have been used to measure pose for anatomical structures [ref force glove, measurand].

Direct field-sensing trackers rely on sensors that measure gravitational or magnetic fields to determine position and orientation. Magnetic trackers were introduced by Polhemus Inc. and, later, by Ascension Technology Corporation [Raab et al., 1979] [Ascension Corp., 1990]. Magnetic trackers have a transmitter, which produces an AC or a pulsed DC magnetic field, and receivers with three orthogonal coils. The current induced in each coil is used to determine the position and orientation of the receiver. Recently, magnetic tracking systems have become “wireless”, in that the receivers are attached to portable, RF activation units [Polhemus Inc., 2000] [Ascension Corp., 2001]. The accuracy and precision of magnetic trackers are drastically affected by electromagnetic interference. However, magnetic trackers continue to be widely used because of their low cost, light weight, and compactness.

Hybrid tracking systems use a combination of operation principles, such as time-frequency measurements combined with inertial sensing. While hybrid technologies increase the complexity of a tracking system (and likely its cost), they are adopted either to access vari-

ables that one technology cannot easily provide (relative and absolute measurements), or to make exhaustive measurements. In the latter case, when associated with filtering and predictive techniques, sensor fusion techniques are used to associate incomplete data sets coming from different sensor types. Examples of hybrid trackers are [State et al., 1996], [Neumann et al., 1999], and [Intersense, Inc., 2000].

## 1.2 Motivation

A main challenge in virtual environments is the accurate tracking of objects within the environment. There are many tracking techniques that are currently used within virtual environments. A current technique is the placement of markers on the objects of interest within the environment and using specialized hardware to recognize them.

An issue when implementing a marker-based tracking approach is determining the configuration of the tracking probe, which is a collection of markers placed on the object to be tracked. For tracking simple motion of a rigid object with constant curvature (implying no changes in concavity), a minimum number of markers placed in a non-collinear, yet arbitrary fashion is sufficient. But, tracking applications involving anything other than the previously mentioned qualifications require a larger number of markers in specific locations. For example, accurately tracking a translating, rigid cube requires only the minimum number of markers on a single side, while tracking the motion of fingers on a hand will require a larger number of carefully placed markers.

Another issue in marker-based tracking is determining the accuracy achievable for a given probe topology. A statistical relationship exists between the markers on a probe and the probe performance. However, to our knowledge, there has never been a thorough investigation of where to place markers to achieve a desired result. It would be beneficial for practitioners to know the effects of changing the probe topology beforehand and to incorporate this knowledge into the design process.

### 1.3 Research Summary

The purpose of the research is to present a framework for conformal tracking in virtual environments. Conformal tracking refers to the creation of tracking probes being constructed according to the shape of the objects of interest (i.e., conforming) and environmental constraints. Our hypothesis is that a framework can be developed to design conformal tracking probes that meet user-defined criteria while minimizing the pose error.

Traditionally, marker-based tracking has implied the use of optical tracking methods. We expand the definition of a marker-based tracking system to include any tracking system that uses multiple, distinct markers to determine the position and orientation of the object(s) tracked. Furthermore, we extend the definition of marker to include features, which are recognizable by the tracker, that indicate the location of objects within the range of the tracker.

The framework presented provides a method for designing rigid, conformal, marker-based tracking probes, predicting the performance of these probes, and quantifying the performance of rigid, marker-based tracking probes. All results are expressed in terms of visual space (i.e., distances instead of pixels). The research is valid for systems in which multiple markers are required to determine the position and orientation of the tracked objects. In addition, the presented research is limited to static analyses of tracking probes.

### 1.4 Dissertation Outline

The rest of the dissertation is organized in the following manner: Chapter 2 gives an overview of virtual environment tracking technology and describes related work in the areas of marker-based tracking, pose estimation, and the quantification of tracking errors.

Chapter 3 details the steps of the framework for conformal tracking. The steps are presented in a flowchart and formally defined.

Chapter 4 describes a pose estimation procedure based upon principal component analysis. We also detail a technique for tracking probes with markers that are not all simultaneously detected by the tracker. We then discuss metrics for quantifying probe performance.

Chapter 5 presents a mathematical framework for propagating the effects of noise to the pose estimation procedure. We also examine the effects of the various environmental factors associated with creating and utilizing a marker-based tracking probe.

Chapter 6 details application of the framework to simple tracking probes. In particular, we examine a planar six-marker probe, a concentric eight-marker probe, and a spherical head tracking probe. We begin by discussing computer simulation of tracking probes and the pose estimation process. Next, experimental results from the tracking probes are presented. Lastly, we detail the application of the conformal probe design process to the creation of an integrated, head-mounted display tracking probe.

Finally, in Chapter 7 we summarize the results and the contributions of the work. We also examine the implications of the framework presented as it applies to ongoing research efforts.

## CHAPTER 2: RELATED WORK

There have been numerous research efforts in virtual environment tracking. The major contributions we shall highlight are marker-based tracking algorithms, marker-based pose estimation algorithms, and research on determining pose errors for tracking in virtual environments.

### 2.1 Marker-Based Tracking Algorithms

Marker-based trackers can be categorized as active marker systems or passive marker systems. In Section 1.3, we extended the definition of marker-based tracking. However, in the past, marker-based tracking has implied the use of optical, spatial scan tracking methods. Therefore, we shall discuss optical marker-based tracking or hybrid systems which utilize optical methods.

The first virtual environment also included ultrasonic tracking technology [Sutherland, 1968]. In the system, each emitter produced a continuous sound wave at a specific frequency. The receivers each detected a single sound wave and compared its phase to that of a reference signal. A displacement of the target from one measure to another produced a modification of the phases that indicated the relative motion of the emitters with respect to the receivers. After three emitters had been localized, the orientation and position of the target was calculated.

An early research effort was the creation of a videometric ceiling tracking system [Wang et al., 1990]. The technique employs several cameras placed on the head of a user to detect infrared LEDs mounted on the ceiling. A mathematical technique based on photogrammetry is used to recover the position and the orientation of a target [Azuma and Ward, 1991]. The tracker had an update rate of 200Hz and was accurate to 2mm in position and 0.1 degrees in orientation. A scaleable version of the tracker was later achieved [Ward et al., 1992].

As part of a system for surgical augmented reality, Mellor introduced a marker-based method for tracking that requires only a single image [Mellor, 1995]. This system was able to recover depth information monocularly and used circular markers. The tracking algorithm uses a set of

three perspective imaging equations for four or fewer landmarks and least squares estimation for larger numbers of landmarks. The tracker is accurate to a fraction of a pixel.

Uenohara and Kanade implemented a method to dynamically track rigid objects utilizing the video output from the cameras of a video see-through HMD [Uenohara and Kanade, 1995]. The tracking method uses template matching to detect at least four markers. Pose recovery is accomplished with a recursive optimization, based on Newton's method, and geometric constraints. The method is capable of tracking at 30Hz.

Ravella et al. developed a tracking method based on features present in the object of interest [Ravela et al., 1995]. The video-based algorithm used intensity maps and edge information in each image to identify the tracked object features. By applying a steerable filter and image correlation, the method was able to track objects across rotation at 8Hz.

A hybrid tracking system able to withstand occlusion was developed by State, et al. [State et al., 1996]. This system for registering real and synthetic objects used marker-based optical tracking for accuracy in conjunction with magnetic tracking for robustness. The magnetic tracking component is calibrated "on-the-fly" so that registration is not lost when markers become occluded. Registration errors for the system were less than one pixel. There were no results reported on errors in visual space.

More recently, the capabilities of the tracking system created by Wang et al. have been expanded by utilizing sensor data immediately when it is obtained [Welch and Bishop, 1997]. This utilization, referred to as Single Constraint At-A-Time (SCAAT) tracking, takes one measurement at a time and updates the tracking system as opposed to waiting for a batch of data. The state of the system is then sequentially updated using an Extended Kalman Filter (EKF). The tracker, now a commercial product from 3rd Tech, Inc., is accurate to 0.5mm in position and 0.1 degrees in orientation.

Foxlin et. al combined inertial tracking with ultrasonic time-of-flight measurements to produce a scalable tracker [Foxlin et al., 1998]. The inertial sensor gives orientation measurements for the



system. The ultrasonic sensors are used to correct drift within the inertial sensor and provide 3D position information. The tracking algorithm for the ultrasonic rangefinding is based upon the SCAAT algorithm [Welch and Bishop, 1997]. However, the system only requires three range measurements and uses fewer ceiling mounted acoustic emitters due to the wide field of sound detection from microphones. An EKF is also used to provide tracking estimates. The tracker is commercially available from Intersense, Inc. and is accurate to 3.4mm.

Neumann et al. developed a hybrid tracking method for use in outdoor AR systems [Neumann et al., 1999]. The system integrated inertial and video-based tracking methods. The inertial component was used to provide sourceless orientation measurements and to predict the user's head position. Video-based methods were then added to correct for inertial drift. The tracker has a registration accuracy of less than 10 pixels over 500 video frames. No results were reported on errors in visual space.

Park and Neumann also introduced two algorithms for tracking in augmented reality to increase the robustness of 3-point pose solutions (three non-coplanar markers used to estimate pose) and iterative pose estimation methods [Park et al., 1999]. To improve 3-point algorithms, a real-time approximation of Huber's M-estimator [Huber, 1981] was implemented to reduce the effect of outliers among the data. A linear Kalman filter was also used for temporal data smoothing. Improvement to iterative pose estimation was accomplished with an iterated EKF, which is a variation on the SCAAT algorithm [Welch and Bishop, 1997]. The system has an update rate of 14Hz and has a registration accuracy to 1 pixel with the 3-point improvements and to 0.52 pixels with the iterated EKF. Error results for visual space were not reported.

Kato and Billinghurst implemented marker-based tracking within a video based augmented reality system for video conferencing, now known as the AR Toolkit [Kato and Billinghurst, 1999]. The markers were squares whose sizes were known a priori. Template matching is utilized to identify markers and perspective projection matrix techniques are used to recover pose. At 0.6m,

the tracker accuracy ranges from 5mm when the markers are perpendicular to the camera to 30mm when the markers are tilted 85 degrees away from the camera.

Edwards et al. utilized augmented reality for overlaying MRI images during a microscopic surgical procedure [Edwards et al., 2000]. Within the application, infrared LEDs were used to locate the microscope and the patient's skull, while passive markers were used with a specially mounted device to track the patient within the microscope field of view. The application was able to achieve registration on the order of 0.5mm for physical phantoms, 1.0mm for bones, and 4mm on target surgical structures.

Zhang and Navab developed a marker-based tracking algorithm for use in a maintenance assistance application [Zhang and Navab, 2000]. The authors compare pose estimation approaches based on homography (at least four coplanar features required) and three markers without a coplanar restriction. The results were that the 3-point algorithm worked better for smaller areas of interest and the homography algorithm worked better for larger areas. This result may be interpreted to mean that tasks which are closer at hand require more depth detail (perspective projection important) than tasks which are performed at a great distance (size cues are more important).

## 2.2 Pose Estimation Algorithms

There are many techniques available for pose estimation. In general, the techniques are robust, using minimization of a least-squares error statistic. In this section, we chose to highlight the techniques that are most relevant to the work presented here; that is methods that utilize individual marker data to form an estimate of pose.

An early effort from the biomechanics literature is the method proposed by Spoor and Veldpaus [Spoor and Veldpaus, 1980]. This pose estimation technique uses Lagrangian multipliers to minimize the sum of squared errors between two sets of marker coordinates. By minimizing the sum of squared errors, the rotation and translation between the two sets is determined. The technique also accounts for uncertainties in the marker positions.

A popular pose estimation method is from Arun, Huan, and Blostein [Arun et al., 1987]. Similar to [Spoor and Veldpauw, 1980], a minimization of the sum of squared errors is performed. However, the method to find the rotation and translation to minimize the error uses Singular Value Decomposition (SVD). By using SVD, the technique is very robust and computationally fast. Moreover, it allows for solutions in case where the data may be ill-formed, such as when markers are close to being collinear.

A similar pose estimation procedure is the method presented by Horn [Horn, 1987]. Horn's method uses the covariance between markers expressed in two different frames to form a 4x4 matrix of relationships. The largest eigenvector from this matrix is then taken to be a unit quaternion. The property of this quaternion is it represents the rotation that minimizes the sum of squared errors between the markers in the two frames.

Another popular pose estimation procedure is [Haralick et al., 1989]. In this work, the markers are iteratively weighted as to reduce the effects of outliers on the pose estimation process. In addition, the work contains numerical simulations of the pose estimation process. The simulations present some of the first rigorous clues that noise adversely affects pose estimation. Moreover, in using the proposed pose estimation method, the number of markers used can enhance the accuracy of pose estimation. The procedure is applied to 2-D pose estimation, 3-D pose estimation, and 2-D perspective projection 3-D pose estimation.

### 2.3 Previous Research in Pose Error Determination

There is also a significant body of work that exists on quantifying the different types of errors present within tracking systems and their effects on pose determination. We limit the scope of our survey of this area to research whose aim is determining errors in pose determination related to the topology of marker distributions.

The common statistical method for error propagation involves computation of partial derivatives based upon a functional relationship between the desired quantity (to which the propagated

error will be related) and the independent variable. Provided a clearly defined functional relationship exists and computation of the derivatives is tractable, this is the method of choice. However, even in more complex cases, the general principle is still applicable.

A closely related technique for spatial error propagation is dilution of precision. Often related with the Global Positioning System (GPS), dilution of precision is directly related to statistical error propagation. The result of this procedure is a covariance matrix with the amounts of error in  $x,y,z$  and in rotations about the principal axes. A requirement is that partial derivatives would have to be computed for the quantities. The overall measure of goodness used is the Frobenius norm of the covariance matrix.

Relating to the statistical method for error propagation, Woltring et al examined the effects of marker errors on pose estimation and provided a maximum error statistic to predict the pose error of a given tracking probe topology [Woltring et al., 1985]. The error statistic derived used first-order errors within its error computation. This approach dealt strictly with the case when the markers were symmetrically distributed with respect to the probe origin. In addition, the probe origin was coincident with the marker centroid (e.g. a tetrahedron or a cube).

In an extension of the work in [Woltring et al., 1985], Morris & Donath quantified the cumulative effects of multiple error sources, including the effects of algorithmic errors and dynamic target array deformation errors [Morris and Donath, 1993]. Based upon the work presented in [Woltring et al., 1985], a modified maximum error statistic was presented for determining the pose error for a given tracking probe topology. The research was valid for marker distributions that lay on a sphere and had the center of the sphere as the probe origin (e.g. a square, pyramid, tetrahedron, etc.) .

In a context related to virtual environments, Vogt et al implemented a method for designing tracking probes using a Monte Carlo simulation technique [Vogt et al., 2002]. The design methodology minimized the jitter error associated with the tracking probe, using the probe radius, marker heights, and number of markers as input variables. Because Tsai's calibration technique

[Tsai, 1987] was utilized to determine the pose of the tracking probe, the method required probe topologies with at least seven, simultaneously detected markers.

Allen and Welch [Allen and Welch, 2004] have implemented a system to dynamically model and predict the performance of tracking systems. The ARTEMIS system uses a state-space based approach to model the dynamics of the tracker and the probe within the tracking volume. Using a discrete, algebraic Riccati equation, the algorithm is able to determine the steady-state performance of a tracker and tracking probe at various locations in space.

The approach presented in this work expands upon the previously mentioned efforts by presenting a theoretical framework for determining pose error that is valid for any probe topology with at least three, non-collinear markers. Moreover, the error sources for the location of each marker are considered as an ensemble, and the error on each marker considered is thus a combination of jitter, tracker bias, probe deformations, and the probe topology. Combined with previous work regarding methods for marker placement [Davis et al., 2002] [Hamza-Lup et al., 2002], this research represents a starting point for a general framework for designing marker based tracking probes.

## CHAPTER 3: CONFORMAL PROBE DESIGN

When implementing a marker-based tracking approach, the configuration of the tracking probe, which is a collection of markers placed on the object to be tracked, must be determined. We refer to the process of determining the number and arrangement of markers on a tracking probe as probe design. Conformal probe design extends the concept of marker arrangement to consideration of the tracking environment and the application requirements to optimize probe performance. By conforming to the constraints placed by the tracking environment and the application, a probe can be designed heuristically that performs well. However, it would be more practical to establish an iterative design procedure which accounts for the interrelation between environmental and application factors.

Practitioners usually take a bottom-up approach to building applications or systems in virtual environments, assembling pieces to create a larger whole. The result is often a "Frankenstein" system that may accomplish the desired task, but is inelegant and causes users to conform to it. As humans, our systems and applications should conform to us, not vice versa. Moreover, an elegant design has advantages that may include improved ergonomics and the potential for enhanced performance. Thus, an added benefit of this framework is to promote a top-down design method for marker-based tracking, which allows exploration of the design space and ultimately designing to the specifications of the application.

In this chapter, we shall first outline the steps of an iterative, conformal probe design process. We then detail the major steps of the process. The first area discussed is performance goals for tracking. Next, we examine environmental and application factors affecting the conformal probe design and the tradeoffs associated with each. Following this, we introduce methods for arranging markers upon the object to be tracked. Finally, we define metrics for quantifying probe performance.

### 3.1 A Procedure for Conformal Probe Design

The procedure for designing conformal probes is by necessity an iterative process. A starting point is determined with necessary environmental parameters specified. From there, one or more of the constraints are varied until a satisfactory tracking probe topology is obtained. If the topology meets the application demands, then it would be constructed and tested. If not, the environmental parameters may have to be adjusted to achieve a satisfactory probe design before construction and testing.

The first step in the design process is to specify the pertinent environmental variables. As the name implies, these variables are specific to the tracker and the application. The variables include the tracker FOV, accuracy, and precision, the marker cone of emission (if using active markers), the area within the tracker FOV where the probe will be utilized, and the object to be tracked. The designer must also consider the type of tracking probe being developed, meaning a determination must be made as to whether the markers will be placed directly upon the object to be tracked, or indirectly through a separate structure attached to the object of interest.

Next, one must determine the desired performance specifications. The specifications may include the field of regard, the accuracy, the precision, the size, and the number of markers of the final tracking probe. Depending upon the flexibility in obtaining a solution, some specifications may be left for determination within the design iteration. The last step before beginning the iterative phase is to create or obtain a 3D model of the probe body, composed of triangular polygons.

The iterative phase of the design process begins with mapping the markers on the probe. The mapping may be done using the algorithms presented in Section 3.3.2 or using another systematic method. After the mapping is completed, the field of regard of the tracking probe must be checked. This is accomplished by simulating the rotation of the probe in 0.5 degree increments through the desired field of regard. At each pose, we determine if at least  $K$  markers can be detected by the tracker. If this requirement is not met, there are three options available. The first option is to retry the marker mapping. This is the first option because the mapping process uses optimization

procedures which may differ slightly because of randomization. If this is unsuccessful, the designer may investigate adding more markers, reapplying the mapping algorithm, and re-checking the field of regard. If the desired field of regard is still not attained, another method would be to make the tracking probe smaller, reapplying the mapping algorithm, and re-checking the field of regard. If the desired field of regard has not been attained, then the practitioner may have to consider relaxing some environmental constraints.

After the field of regard requirement has been met, the next step is to achieve the desired accuracy and precision. The accuracy of a tracking probe may be determined by simulating the process of measuring the probe pose, changing the pose by a known amount, and remeasuring the pose. The average difference between the two poses will give a measure of the accuracy of the tracking probe. In simulating the process of measuring the probe pose, we utilize a custom algorithm for propagating the errors from noisy marker data (detailed in Chapter 5). Moreover, we can isolate the probe size, marker noise, and number of markers and examine their effect upon the probe accuracy. The precision of the tracking probe is determined directly from the standard deviation of the simulated marker data. If the probe accuracy is not high enough, then the probe size must be increased (motivating a field of regard re-check and an accuracy re-check) or the number of markers must be increased, causing the probe to move back to the mapping process. The probe precision is directly related to the tracker precision and the precision of marker placement on the probe during final construction, and is therefore not directly affected by the design process.

After the desired accuracy level has been achieved, the final stage of the iterative phase is to determine if the number of markers used and the size of the probe are satisfactory. If there are too many markers on the probe, then a re-design is necessary, starting from the marker mapping phase. If the probe size must be constrained, it may be adjusted provided the probe is then re-evaluated for accuracy and field of regard considerations.

Once the iterative stages have been completed, the newly designed probe must be constructed. The markers should be placed with care, as the precision with which the markers are placed has



direct bearing on the overall precision. The tracking probe must then be calibrated. The calibration of a tracking probe involves establishing the local coordinate frame and computing the probe offset (if needed), which is a small translation of the local coordinate frame to ensure robust measurements. The final step in the conformal design procedure is to assess the performance of the tracking probe by measuring its pose accuracy and precision in the laboratory.

A flowchart for the conformal probe design process is shown in Figure 3.1 with the iterative steps shown in Figure 3.2.

### 3.2 Environmental and Application Factors

The tracking environment is specific to each tracking system and virtual environment application. Within the environment there are several factors which could affect tracking performance. The environmental factors which affect marker-based tracking are the tracking accuracy, the shape and size of the tracked object, and the number of markers. Additionally, the application factors which will affect tracking performance are the field of regard of the tracking probe as well as the number of markers utilized. We shall now briefly summarize each factor.

The tracker accuracy and precision are the most critical factors affecting tracking probe performance. The accuracy and precision of a tracking probe is directly related to the accuracy and precision of the tracker, a relationship we explore in detail in Section 5.2.1. If the tracker cannot provide accurate and precise measurements, then the tracking probe will perform poorly. Importantly, the level of accuracy required is relative to the application. For example, an accuracy of 0.5m is unreasonable for tracking human motion, but would be acceptable (even quite good) for GPS. In addition, a tracker may have varying levels of accuracy depending upon where the probe is placed within its field of view.

The shape and size of the tracked object is another environmental factor to consider when designing marker-based tracking probes. Ultimately, we strive to place the markers to allow adequate freedom of motion for the tracked object within the tracking volume and to utilize the fewest num-

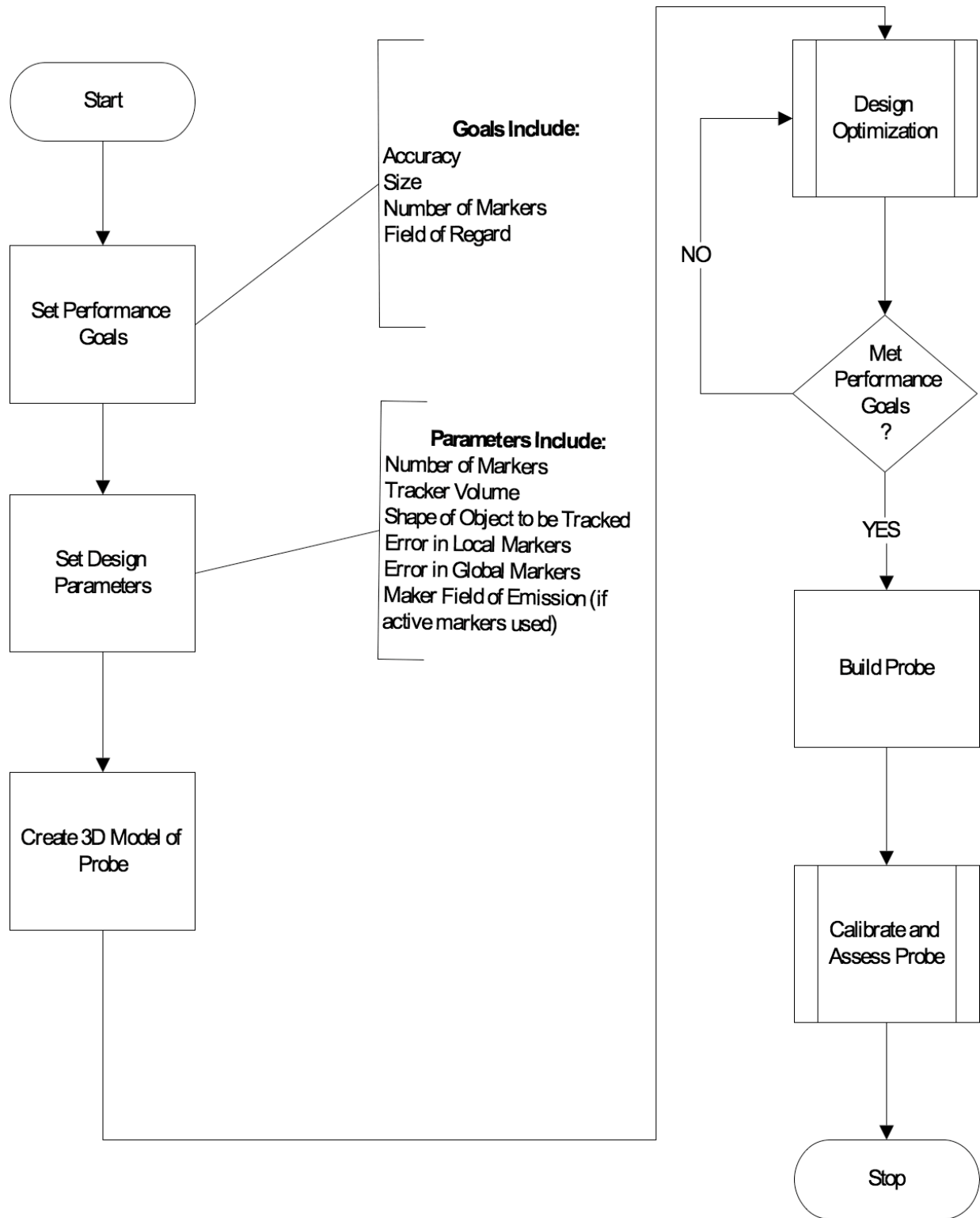


Figure 3.1: An Algorithm for Conformal Probe Design

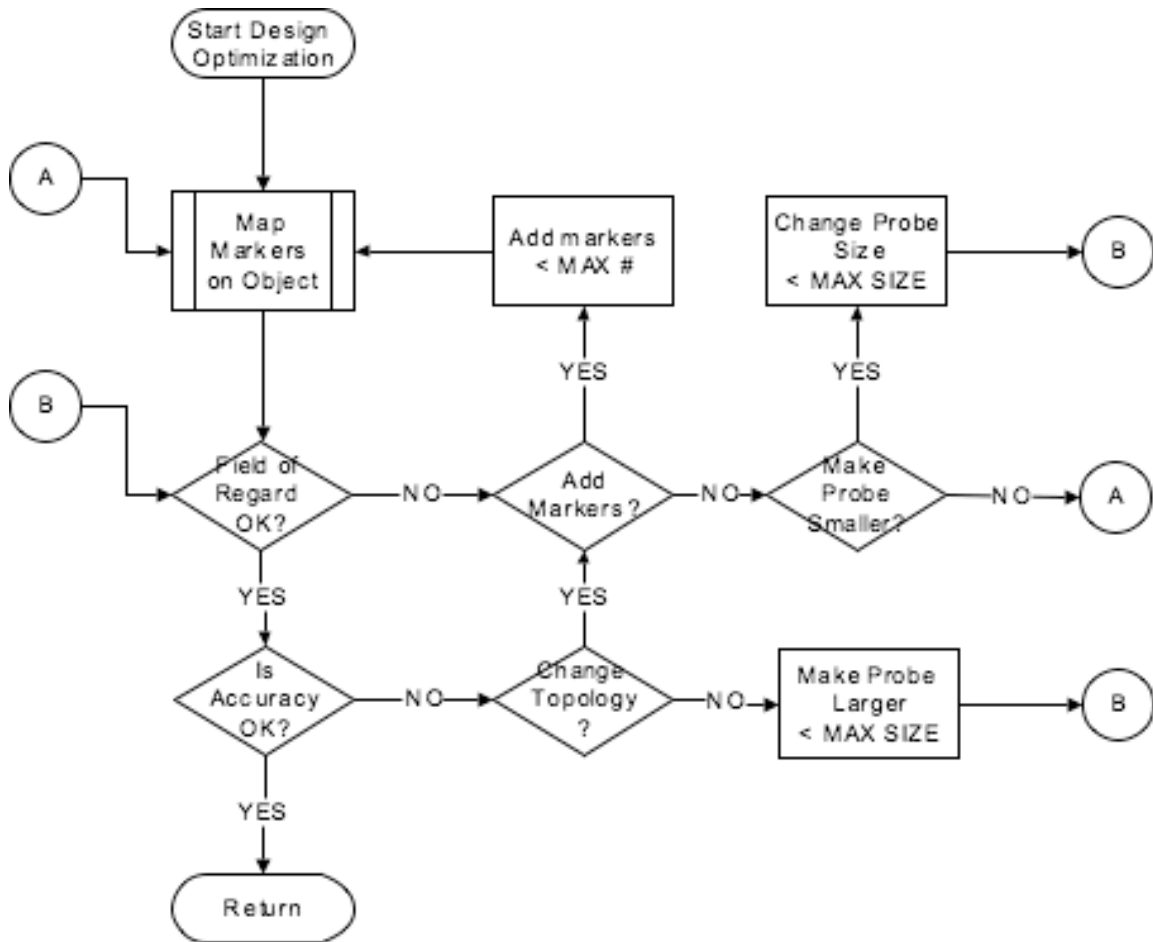


Figure 3.2: Iterative Steps of Conformal Probe Design

ber of markers possible. These goals are affected by the shape of the tracked object, and, thereby, the tracking probe. Moreover, the size of the tracked object will dictate how far apart markers can be placed. This is important because the sensitivity of a tracking probe to changes in orientation increases as the spatial extent of its markers increases.

The field of regard is the angular extent through which a probe that is within the tracking volume can be rotated and still remain detected by the tracking system. The minimum field of regard requirement is determined by the application. For example, if an application requires that an object translating with a fixed orientation be tracked, then a large field of regard would not be necessary for the tracking probe. However, if we were to track the head of a user in an immersive walk-through, the probe field of regard would need to be significantly larger due to the possible range of head motions. Of the factors affecting conformal probe design, the field of regard is the most critical because of its relation to the other factors; to change the field of regard, the number of markers on the probe must be changed or the markers on the probe must be rearranged, i.e., change the probe topology.

The number of markers on the probe is both an environmental and application factor. It is an environmental factor because of marker constraints that are imposed by the tracker. For example, an optical tracker may require the detection of at least four, non-collinear markers to determine the pose of a tracking probe. Moreover, a tracker may be more accurate with more markers, depending upon the algorithm(s) it uses for pose estimation. The number of markers is also an application factor because there may be speed and accuracy issues associated with the application. For example, if a tracking system uses sequentially activated markers, increasing the number of markers used will lower the tracking frame rate. Also, the latency of the tracker measurements may be affected due to increased amounts of time required to process the data from more markers, which can lead to registration inaccuracies or cause user motion to become uncoupled from computer-generated stimuli.

### 3.3 Marker Placement Methods

To maximize the field of regard for a tracking probe while minimizing the number of markers, the markers must be uniformly distributed in angle, meaning the overlap of the cone of emission of each marker covers the same solid angle when projected on a sphere surrounding the tracking probe. In addition, a sufficient number of markers must be used such that the minimum marker constraint for the tracking system is always met throughout the desired motion of the probe.

The first challenge is determining how to generally distribute the markers for any given probe shape. A group of  $N$  markers can be uniformly distributed on a surface by dividing it into  $N$  regions of uniform surface area (such as in regular tessellation). However, this may be extremely difficult or impossible due to the probe shape. Another possible technique for distributing markers is to use the Platonic solids to place markers directly. The Platonic solids are, however, limited to spheres (or spherical equivalents) and to 4, 6, 8, 12 or 20 markers. Additionally, the Gaussian quadrature method has been previously proposed and implemented to uniformly distribute an arbitrary number of markers. A Gaussian quadrature formula is a numerical integration formula, for functions on the sphere, which is exact for all spherical harmonics with  $Y_l^m(\Theta, \phi)$  with  $l \leq L$  [Sobolev, 1962]. The points on the sphere where the function to be integrated is sampled are therefore uniformly distributed. The number of sample points is determined by  $L$ , and increases as  $L$  increases. However, the Gaussian quadrature formulae do not generalize to other shapes.

Thus, a robust approach to use when placing markers on arbitrarily shaped rigid probes is global optimization. In this section, we present a modified version of the simulated annealing algorithm for uniformly distributing markers on spherical objects. The marker distribution algorithm may also be extended to other shapes that can be defined via parametric equations. We then present two conformal mapping algorithms for placing markers on arbitrary objects using our modified simulated annealing algorithm.

### 3.3.1 The Modified Simulated Annealing Algorithm

The goal of an optimization procedure is to find a combination of variables that minimize or maximize a given quantity. The function that defines the relationship between the variables and the desired quantity is known as an objective function or a cost function. For mapping markers on a tracking probe, the cost function must express the relationship between the positions of each marker and the overall distribution of the collection of markers. If the markers are modeled as point charges, then a functional relationship can be inferred from the concept of potential energy. From physics, we know that a pair of point charges in free space have a potential energy proportional to the inverse of the distance between them. Furthermore, the potential energy of a collection of point charges increases as the sum of the pairwise energy interactions experienced by each point charge. Therefore, the simplified cost function utilized is

$$E = \sum_{i=1}^N \sum_{j=i+1}^N \frac{1}{r_{ij}}, \quad (3.1)$$

where  $N$  is the number of markers and  $r_{ij}$  is the Cartesian distance between the  $i$ th and  $j$ th marker.

By definition, minimizing  $E$  will maximize the distance between the markers, subject to the constraints imposed by the dimensions of the object upon which the markers will be mapped. To minimize the cost function and thereby uniformly distribute markers on a tracking probe, we choose to apply the method of simulated annealing. The method of simulated annealing (SA) was first used in an algorithm designed to simulate the cooling of a material in a heat bath by Metropolis in 1953 [Metropolis et al., 1953]. In 1983, Kirkpatrick suggested that this type of simulation could be used more generally to search the feasible solutions of an optimization problem, with the objective of converging to an optimal solution [Kirkpatrick et al., 1983] [Dowsland, 1993].

The SA optimization process is analogous to the metallurgical process of annealing, which is used to increase the strength of metals by minimizing the potential energy between molecules. In this process, a metal structure is heated past its melting point, allowing the molecules to move

freely throughout and increasing the overall potential energy. The metal is then slowly cooled. Although there is a probability that a given molecule may move to a higher energy state (based upon physical laws and related by Boltzmann's constant), the overall trend is for the molecules to move to lower energy states, reducing the potential energy. Moreover, the probability of a molecule moving to a higher energy state is directly proportional to the current temperature of the metal, meaning that the cooler the metal gets, the less likely that its molecules will enter higher energy states. Eventually, the metal returns to a solid, locking the structure of the metal. If the process was performed well, that is, the metal was made hot enough and the cooling was done slowly enough, the molecules of the metal will be in the minimum (or very close to minimum) potential energy state, resulting in a very strong metallic structure. If the metal was not heated enough or cooled slowly enough, the resulting metal will have imperfections, creating weak spots or brittleness.

If we consider the solution space for our optimization as the structure of the metal, we can see certain trends and strategies for SA. The location of the molecules within the metal represents different solutions. The movement of the molecules represents our search of the solution space. The movement is facilitated by higher temperatures. The temperature change is dramatic at higher values and minute at lower values, allowing the system greater fluctuation at higher temperatures while allowing it to "settle down" as it reaches lower temperatures. This results in a larger area of the solution space being searched at the beginning of the algorithm, while the solution is slowly refined. If our system is "heated" and "cooled" appropriately, then we will insure an adequate search of the solution space for our problem. By applying a probability of movement to a higher energy state, local minima can be avoided while narrowing the solution space. Finally, the metal coming back to a solid state represents reaching an optimal (or close to optimal) solution.

We now describe the algorithm as applied to the distribution of markers on a sphere, with extensions to the traditional SA algorithm indicated in the text. To start, an initial temperature is chosen and the number of points to be distributed,  $N$ , is determined. The points are placed randomly on a sphere of unit radius. The annealing process begins with the computation of the

starting value of the potential energy of the system,  $E$ , which varies inversely with the distance between all the points on the sphere.

After computing the starting energy of the system using Equation 3.1, an iterative process begins. First, the system temperature is decreased according to the cooling schedule. Next, a point on the sphere is chosen randomly and moved randomly in azimuth and elevation on the surface of the sphere. The current energy of the system is then calculated. If this current energy is less than the previous system energy, then the move is accepted. If the current energy is more than the previous system energy, a probability of accepting the move is generated based upon the current temperature. The equation used for computing the probability of accepting a point movement is given by

$$\begin{aligned} \text{Probability} &= \frac{e^{-\Delta/T}}{1 + e^{-\Delta/T}} \\ &= \frac{1}{e^{\Delta/T} + 1} \end{aligned} \quad (3.2)$$

where  $\Delta$  is the change in energy of the distribution due to the point movement, and  $T$  is the current temperature. A uniform random number between zero and one is generated, determining whether the point is moved. If the random number is less than or equal to the probability in Equation 3.2, the selected point is moved. Otherwise, the point remains stationary. In either case, another point is selected and moved following the same criteria until  $K$  iterations have been performed at the current temperature. After  $K$  iterations, the temperature is decreased and the process is repeated at the next lower temperature. When the system temperature is less than 0.1, the iterative process stops and the system freezes. The final energy value and the distribution of the points are then computed. The markers are mapped to the point locations. A flowchart of the modified SA algorithm is presented in Figure 3.3.

Differences between the SA algorithm presented and that employed in [Kirkpatrick et al., 1983] appear in the implementation of the cooling schedule and the probability expression. A geomet-



ric temperature reduction was chosen instead of an exponentially decreasing cooling schedule, decreasing the system temperature by a fixed percentage of the current temperature after  $K$  iterations.

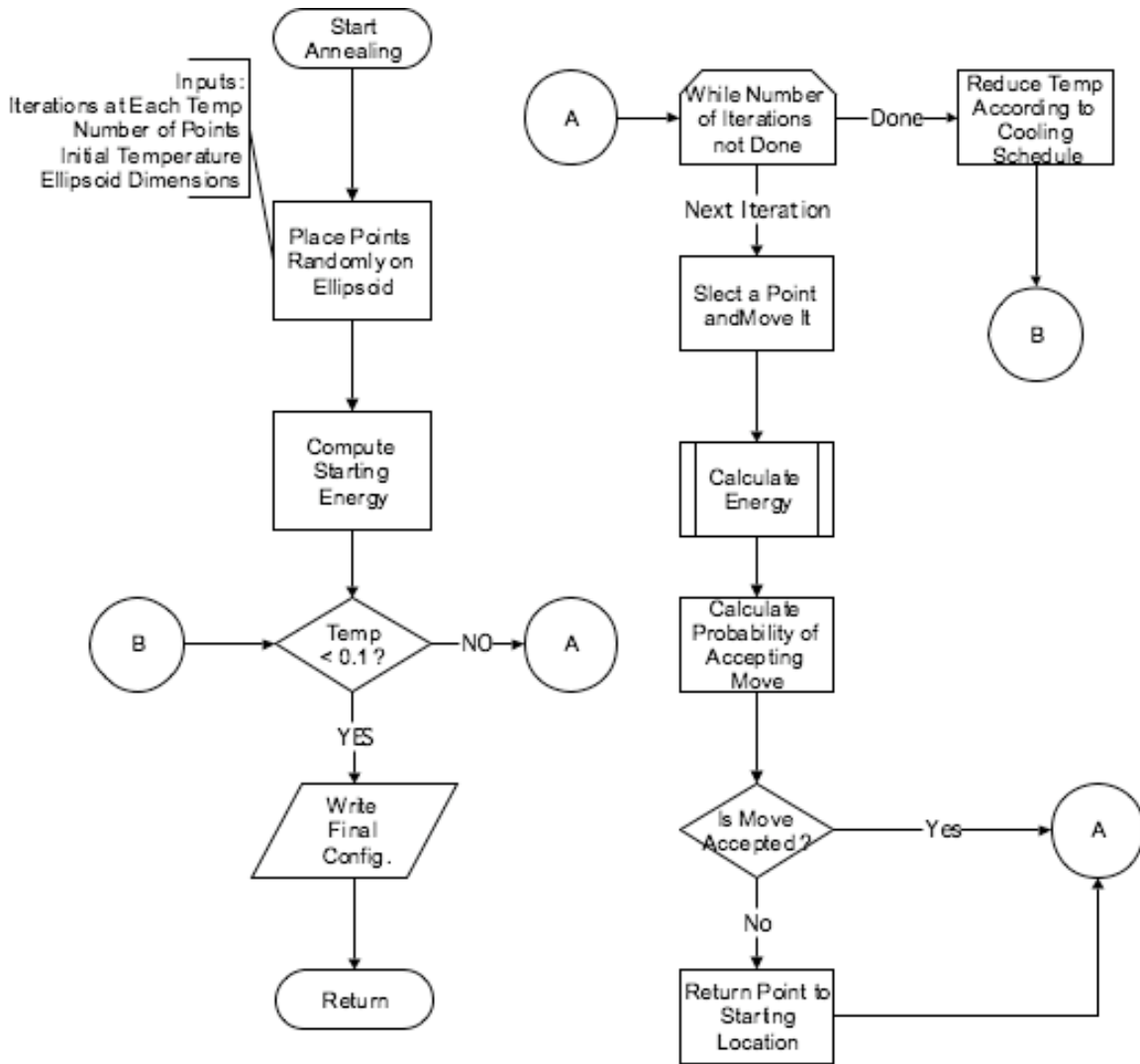


Figure 3.3: The Modified Simulated Annealing Algorithm

The exponential temperature reduction was proposed because of its simplicity. Also, a simplification of the traditional probability expression was employed as shown in Equation 3.2. Here, Boltzman's constant was removed and the expression was rationalized to give values that fall between 0 and 0.99.

To apply the modified SA algorithm to shapes that can be defined parametrically, e.g. an ellipsoid, we use a slightly different cost function. Instead of using a potential energy function based upon Cartesian distances, we base the function upon arc lengths. Thus, Equation 3.1 becomes

$$E = \sum_{i=1}^N \sum_{j=i+1}^N \frac{1}{s_{ij}} , \quad (3.3)$$

where  $s_{ij}$  is the arc length between the  $i$ th and  $j$ th point. This change allows objects with non-uniform curvatures to have marker mappings that maximize the field of regard for the tracking probe.

### 3.3.2 Conformal Mapping for Arbitrary Objects

Conformal mapping is a mathematical technique for transforming one solution space to another using complex variables. Traditional conformal mapping techniques could be applied for placing markers on objects whose shape is defined mathematically or can be easily approximated. However, we generally do not know the shape of the tracking probe and must rely upon other methods. We have implemented two techniques for placing markers on arbitrary 3D objects. The first technique is based upon texture mapping and uses an intermediary surface to place markers. It is used for regular objects, meaning that the object is mostly convex and does not undergo large or frequent changes in concavity. The second approach involves simulating the detection of markers with virtual viewpoints and is used when placing markers on objects that are irregular. Both mapping techniques require a 3D, triangular polygonal model of the object, which can be obtained by digitizing the object.

#### 3.3.2.1 Marker Mapping with Intermediary Surfaces

To solve the problem of mapping markers on arbitrary objects, we can utilize a mapping technique that is applied with an intermediary three-dimensional surface. The technique is similar to texture

mapping in that a region of the intermediary surface (the region being a marker instead of a surface patch) is projected onto the arbitrary object. The intermediary surface used is an ellipsoid, simplifying to a sphere when the ratio of the largest to the smallest eigenvalue of the dispersion matrix of the vertices comprising the arbitrary object is less than ten.

The equation of the ellipsoid is derived from the characteristics of the 3D model of the object. The center of the ellipsoid is chosen to correspond with the centroid of the 3D model. The radii of the ellipsoid are given by the norms of the eigenvectors that characterize the distribution the vertices of the 3D model of the object, the principal axes. The distribution of the vertices is

represented by the dispersion matrix. Given  $N$  vertices, the 3x3 symmetric dispersion matrix,  $\mathbf{D}$ , is defined as

$$\mathbf{D} = \frac{1}{N} \sum_{i=1}^N \tilde{\mathbf{x}}_i \tilde{\mathbf{x}}_i^T \quad (3.4)$$

with

$$\tilde{\mathbf{x}}_i = \mathbf{x}_i - \mathbf{x} \quad , \quad (3.5)$$

where  $\mathbf{x}_i$  is the location of the  $i$ th vertex,  $\mathbf{x}$  is the centroid of the vertices, and  $\tilde{\mathbf{x}}_i$  is a 3x1 vector representing the location of the  $i$ th vertex with respect to the centroid of the vertices. The final intermediary surface is a proportional ellipsoid scaled to encompass the vertices of the 3D model.

After defining the ellipsoid for the intermediary surface, we use the simulated annealing algorithm in Section 3.3.1 to uniformly distribute points that represent marker locations on its surface. The 3D model of the object is then placed within the intermediary surface, with the principal axes and the centroids of each aligned. The points are mapped to the 3D model by casting a ray from each point on the intermediary surface to the centroid of the 3D model. The final location of the markers on the model are determined by tracing these rays to their intersections on the 3D model. A flowchart of the intermediary algorithm is illustrated in Figure 3.4.

To determine the points of intersection on the 3D model, we start with the  $M$  vertices of the model and  $N$  points on the intermediary surface. For each point on the intermediary surface, we compute the equation of the parametric line containing the point and the centroid of the 3D model. We first determine if the triangle is candidate for intersection by computing the dot product of its normal and the vector between the centroid of the model and the point on the intermediary surface. If the dot product is greater than zero, then the angle between the vectors is less than 90 degrees, indicating that the triangle does not face the point on the intermediary surface and can be eliminated from consideration. If the dot product is negative, then we make two comparisons of the line to the set of triangles that comprise the 3D model. We check that the line intersects the plane formed by the triangle vertices and if the intersection point is within the triangle. If these

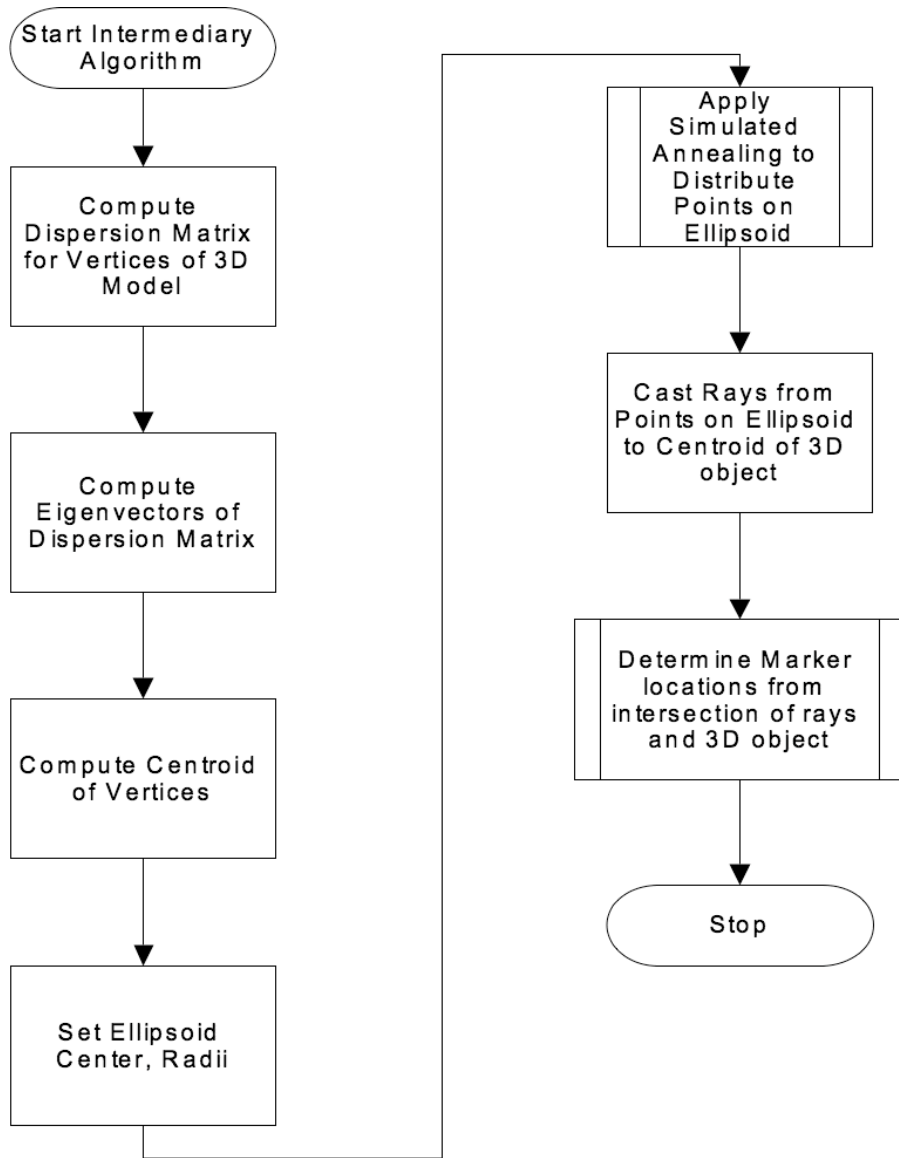


Figure 3.4: The Intermediary Algorithm

conditions are all met, then the point on the intermediary surface is mapped to the triangle under consideration. A flowchart of the method for determining intersection is given in Figure 3.5.

### 3.3.2.2 *Marker Mapping with Virtual Viewpoints*

In general, marker-based trackers must detect at least three markers to determine the pose of a tracked object. Therefore, we attempt to place markers upon an object to meet this requirement. From the perspective of the tracker, the markers are detected or “in view”. Knowing the point of origin and extent of the tracking volume, i.e., the tracker “view”, we can reverse the problem; instead of placing many markers on the object to conform to one “view”, we could place many “views” around the object and determine where to place a marker. We can achieve this reversal by using the 3D model of the object and placing it at the center of an intermediary sphere. The sphere would then have the “views”, which we call virtual viewpoints (or just viewpoints), represented by points optimally distributed on its surface using the method given in Section 3.3.1 or distributed according to the desired field of regard. By counting the number of times a polygon is “seen” by one of the virtual viewpoints, we can determine the optimal placement of markers. The centers of the polygons that are “seen” most often are chosen to become the marker locations.

To create a tracking probe using the viewpoints algorithm, we start by distributing the viewpoints on the intermediary sphere surrounding the 3D model of the object. The intermediary sphere has a radius that is twice the maximum distance between two vertices in the triangular mesh of the 3D model to ensure enclosure. For each viewpoint, we then determine whether a marker placed on any of the triangles in the 3D model could be “seen” by the viewpoint. The determination is based upon the angle between each triangle normal and the vector from the center of the triangle to the viewpoint in question. If the angle is less than half of the field of emission of a marker, then the triangle can be “seen.” If the markers being considered are passive, then the angle must be less than 90 degrees. In either case, the number of viewpoints that can “see” the triangle is incremented by one. This test is applied to all the triangles in the model with respect to the same

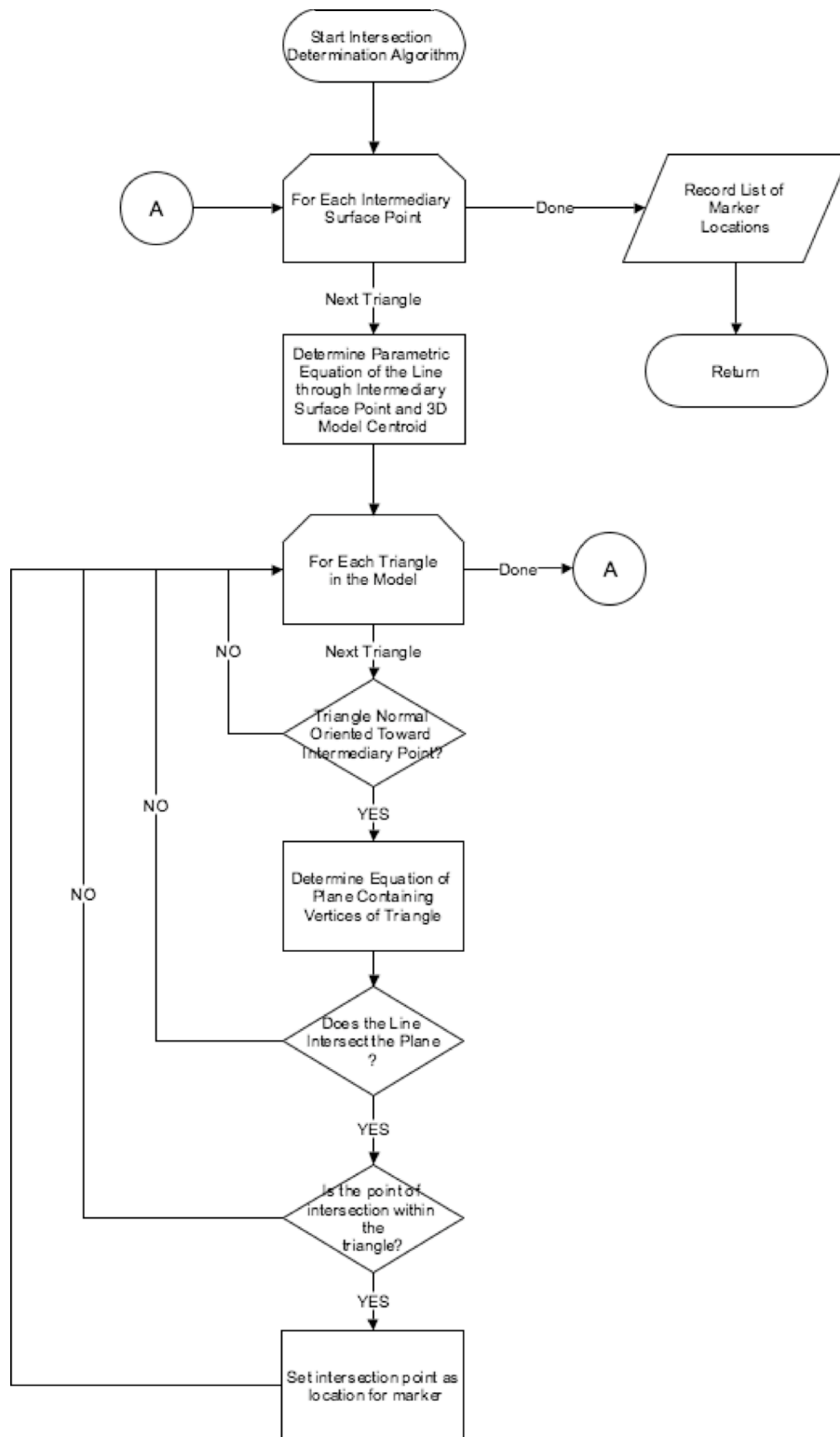


Figure 3.5: The Intersection Determination Algorithm

viewpoint. The process of testing for triangles which can be “seen” is repeated for all viewpoints on the intermediary sphere. Once all the viewpoints have been tested, the triangle which has the highest viewpoint count is selected as a final marker location and is removed from further consideration. Each viewpoint that can “see” this triangle also has its marker count incremented. Once a viewpoint can “see”  $K$  triangles (its marker count equals  $K$ ), it is removed from consideration also. The process for determining how many triangles are seen by the viewpoints is continued until all viewpoints “see”  $K$  triangles. The final marker mapping is then saved and used for further analysis.

A potential problem with the viewpoints algorithm exists when dealing with complex objects, that is, objects that cannot be defined by simple mathematical functions or simple parametric equations. For example, if we were trying to map markers on a 3D “U,” there would be triangles that satisfy the “seen” requirement due to their normals, but would be “hidden” because they were located on a part of the “U” that was obscured by another portion of the letter. This is the situation that occurs if one imagines holding the “U” upright in your hand, rotating it 90 degrees to the right, and trying to view the inner portion of the “U.” The solution to this problem lies in using the parametric equation of the line passing from the viewpoint through the center of the triangle being considered. Once we have determined that a triangle can be “seen” by the viewpoint, we check to see if the line intersects any other triangles. If intersections occur beyond the first triangle, we check the angle criteria for each additional triangle intersected by the line. If we determine that the additional triangles pass the “visibility” test, we determine the point of intersection of the line for these triangles. We then substitute the intersection points for each of the “visible” triangles that intersect the line (including the triangle we started with) into the parametric equation and solve for the linear parameter. The intersection point that gives the smallest linear parameter is inside the triangle that is closest to the viewpoint, and, therefore, the triangle that is actually “seen.” The other triangles that gave false “visibility” tests are removed from further consideration for the given



viewpoint. Flowcharts illustrating the virtual viewpoint algorithm and the method for updating the number of times a triangle is “seen” are given in Figures 3.6 and 3.7, respectively.

The benefits of the virtual viewpoint algorithm compared to marker placement with the intermediary surface algorithm are threefold. First, this algorithm minimizes the number of markers used for the tracking probe. Second, it guarantees that at least  $K$  markers are visible from each viewpoint, where  $K$  is the minimum number of markers that are necessary for different tracking systems. Finally, a tracking probe with a custom, large field of regard can be created by increasing the number of viewpoints and arranging the viewpoints to emphasize a particular direction (e.g. putting the viewpoints at the front of an HMD).

The cost function of the intermediary algorithm is based upon maximizing the distance between neighboring markers, while the viewpoint algorithm maximizes the number of times a triangle is “seen.” The intermediary approach, therefore, yields a configuration that is optimal in terms of the field of regard, while the viewpoint algorithm minimizes the number of markers and achieves the desired field of regard. Combined with the advantages listed in the preceding paragraph, these facts make the viewpoint algorithm the marker mapping technique of choice in most cases.

A drawback of the viewpoint algorithm, however, is its sensitivity to the density of the triangles on the 3D model. For example, a flat surface can be represented with relatively few polygons, while a complex, curved surface requires many polygons. If a complex feature on the 3D model is prominent, then many markers may get mapped in a very small area when one marker may be sufficient. Thus, the intermediary algorithm may provide better results in this case.

### 3.4 Metrics for Probe Performance

The accuracy of a measurement is defined as “the qualitative expression of the closeness of the result of a measurement to the true value of the measurand” [Rabinovich, 1995]. A quantitative description of this concept is absolute error, defined as “the difference between a value of a measurand obtained by a measuring instrument and the true value of the measurand” [Rabinovich, 1995].

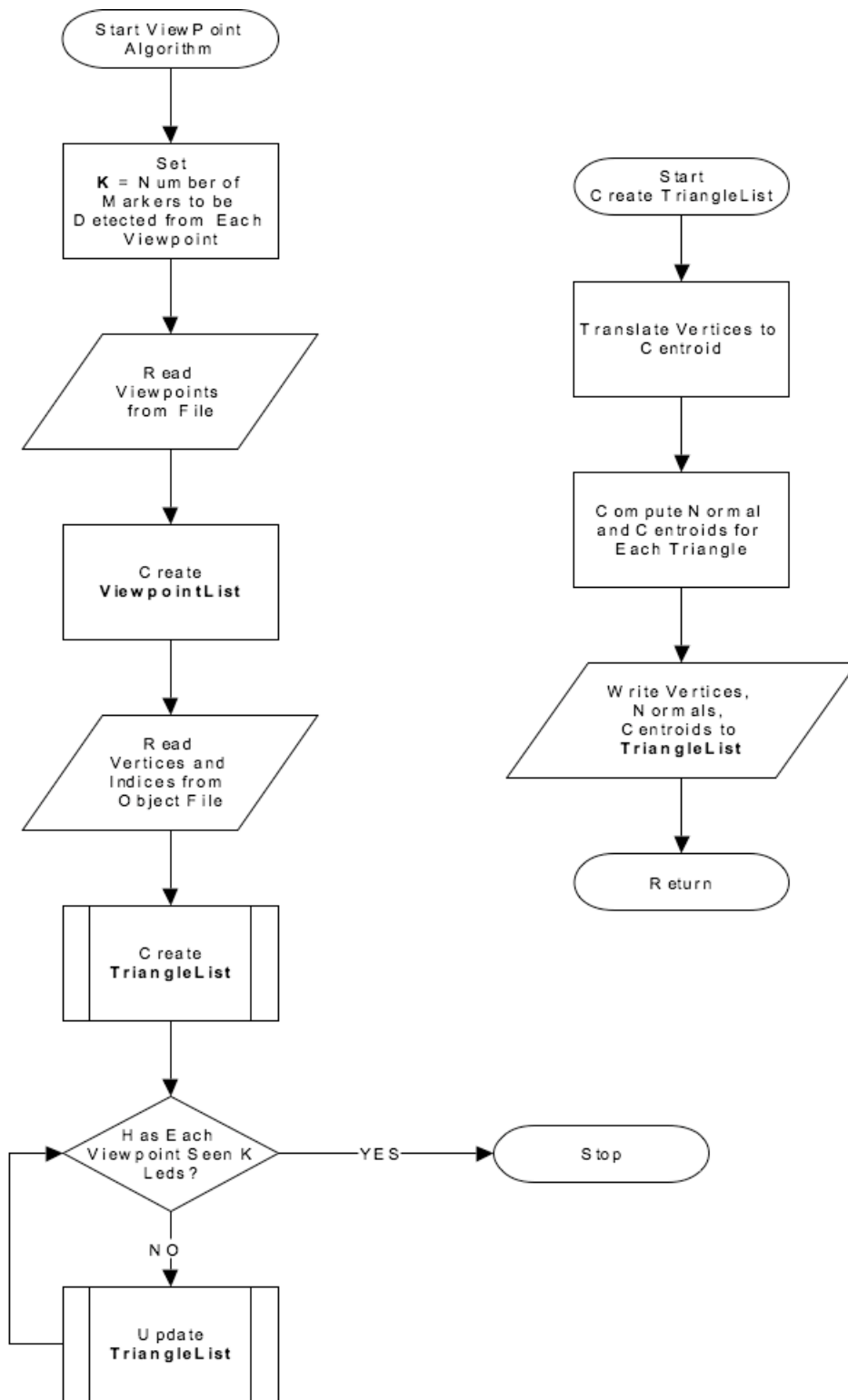


Figure 3.6: The Viewpoint Algorithm

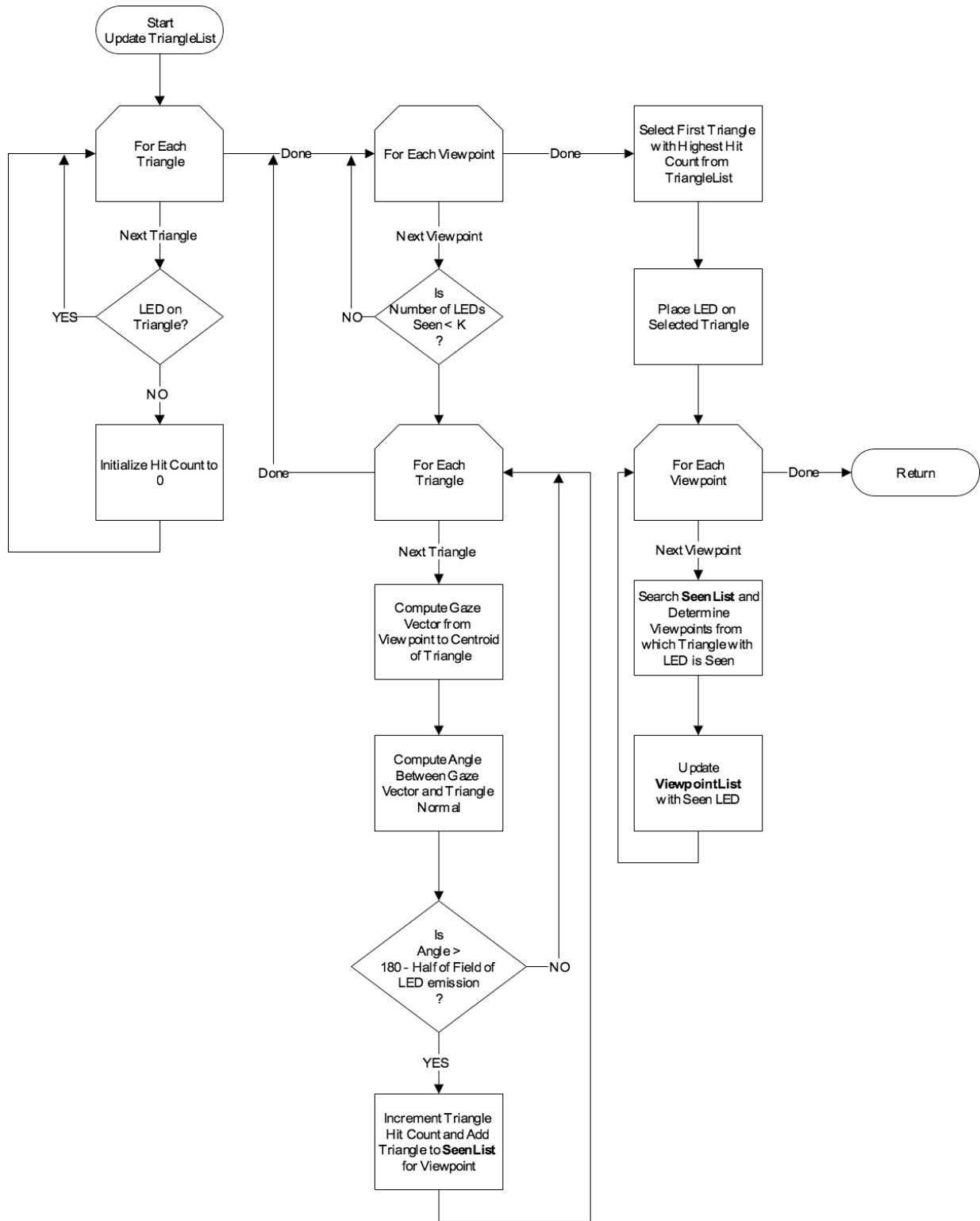


Figure 3.7: Detail of the UpdateTriangles Routine

Resolution, on the other hand, is defined as “the smallest interval between two adjacent values of the output signal of a measuring instrument that can be distinguished” [Rabinovich, 1995]. Furthermore, related to resolution, is the repeatability (precision) of a measurement, which is defined as “the closeness of agreement among a number of consecutive measurements for the same measurand performed under the same operating conditions with the same measuring instruments, over a short period of time” [Rabinovich, 1995].

Armed with these definitions, we can begin to define metrics to quantify the performance of a tracking probe. The metrics of interest include the accuracy, resolution, and precision of pose estimation for a given tracking probe. The field of regard of a tracking probe is also a metric of interest, but is easily determined. The difference in reported and actual pose,  $\Delta P$ , can be expressed in terms of position, orientation, or a combination of both. In position, the difference is expressed as

$$\Delta P = \|T_{reported} - T_{actual}\| , \quad (3.6)$$

where  $T$  is the translation of the tracking probe. In orientation, we calculate the difference as a concatenation of two quaternions, expressed as

$$\Delta P = q_{reported}q_{actual}^{-1} , \quad (3.7)$$

where  $q_{reported}$  is the quaternion representing the measured orientation and  $q_{actual}$  is the expected orientation. To combine position and orientation measures, we use a sum of squared errors expression, applying the rotation and translation of the probe to each marker locations expressed in the local frame of the tracking probe,  $x_i$ . The combined relationship is expressed as

$$\Delta P = \frac{1}{N} \sum_{i=1}^N \|(R_{reported}x_i + T_{reported}) - (R_{actual}x_i + T_{actual})\|^2 , \quad (3.8)$$

where  $N$  is the number of markers on the probe. Practically, the difference in reported and actual pose must be measured in relatively, i.e., from a given starting position, translate the probe by a

fixed amount and measure the difference between what is reported by the tracker and the actual displacement. A similar procedure can be performed for orientation.

The field of regard is the angular extent through which a probe can be rotated and still remain detected by the tracking system. It can be measured by placing the tracking probe upon a rotating stage/platform and observing when the probe is detected and lost. We start by placing the probe in an orientation where it is not detected by the tracker. Then, the stage is rotated slowly. At the moment the probe is detected, we record the quaternion, that expresses the probe orientation,  $q_0$ . The probe is then rotated until it is no longer detected by the tracker. At that instant, the quaternion corresponding to the final probe orientation,  $q_f$ , is recorded. We can determine the quaternion that expresses the transformation between  $q_0$  and  $q_f$  as

$$q_{result} = q_f q_0^{-1} \quad (3.9)$$

extracting the field of regard from the scalar portion of  $q_{result}$ .

At this point, we have defined a procedure for conformal probe design, factors to consider during the design, and metrics for quantifying probe performance. In the following chapter, we detail the process by which the position and orientation of a tracking probe are determined.

## CHAPTER 4: POSE ESTIMATION

Pose estimation is the determination of the position and orientation of an object. In a marker-based tracking scheme, it involves estimating position and orientation based upon the 3D positions of the markers in some fixed, external reference frame.

Many methods exist for pose estimation based on fiduciary markers. Often, the procedure for determining pose is based upon Principal Component Analysis (PCA), sometimes called the Karhunen-Loève transform. The idea of principal component analysis was first proposed by Pearson [Pearson, 1901]. A practical method for discrete computation of the principal components was first presented by Hotelling in 1933 [Hotelling, 1933]. The analogous transform for continuous functions was presented by Karhunen in 1947 [Karhunen, 1947] and Loève in 1948 [Loève, M, 1948] [Sayood, 1996] [de Ridder, 2004].

In this chapter, we introduce a method for pose estimation based on PCA that expands upon the ideas presented in the fundamental literature. We begin the chapter by discussing geometric and PCA-based procedures for determining a rigid coordinate frame, including the case where all the markers on a probe cannot be simultaneously detected. We then discuss methods for PCA-based pose determination. Finally, we discuss procedures of analyzing the errors resulting from the pose estimation procedure.

### 4.1 Determining a Coordinate Frame

The first step of any pose estimation algorithm is to determine the local coordinate system, or local frame, of the tracked object. A marker-based tracking scheme requires at least three non-collinear markers to create a local coordinate frame. The coordinate frame can be created using geometric methods or PCA-based methods.

### 4.1.1 Geometric Frame Determination

Geometric frame determination methods exploit the fixed relationship between the location of markers as expressed in an external coordinate frame, referred to as the global coordinate frame. In the past, we have determined local coordinate frames using vector product relationships.

Given  $K$  markers on the tracking probe, the frame determination method starts by obtaining the position of three non-collinear markers with respect to the tracking system. We denote the markers as  $\mathbf{y}_k$ , with  $k \in [1, K]$ . Each  $\mathbf{y}_k$  is a 3x1 column vector. Assuming the creation of a right-handed,  $xyz$  Cartesian coordinate system using markers 1, 2, and 3, the +x-axis in global space,  $\vec{x}$  is defined as

$$\vec{x} = \frac{\mathbf{y}_2 - \mathbf{y}_1}{\|\mathbf{y}_2 - \mathbf{y}_1\|} . \quad (4.1)$$

The +z-axis in global space,  $\vec{z}$  is defined as

$$\vec{z} = \vec{x} \times \frac{\mathbf{y}_3 - \mathbf{y}_1}{\|\mathbf{y}_3 - \mathbf{y}_1\|} . \quad (4.2)$$

The +y-axis in global space,  $\vec{y}$  is defined as

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

The origin of the axes in global space,  $O_g$ , can be defined as either the centroid of the markers, or as the location of  $\mathbf{y}_1$ . The  $\mathbf{y}_k$  are now expressed relative to  $O_g$ , as

$$\widetilde{\mathbf{y}}_k = \mathbf{y}_k - O_g . \quad (4.4)$$

Finally, the locations of the markers in the local coordinate frame,  $\mathbf{x}_k$ , are expressed as

$$\mathbf{x}_k = \mathbf{V}^T \widetilde{\mathbf{y}}_k, \quad (4.5)$$

where  $\mathbf{V} = [\vec{x}\vec{y}\vec{z}]$

#### 4.1.2 PCA-based Frame Determination

A PCA-based frame determination has several advantages over a geometric frame determination. One advantage is that PCA takes into account the distribution of the markers when forming the coordinate frame. Thus, the axes of the local frame correspond to the symmetries and trends of the marker distribution. Another advantage of PCA versus geometric frame determination is that the covariance matrix of the data will always yield the same set of coordinate axes relative to the marker distribution. This is independent of the order in which the marker locations are specified, as opposed to a geometric determination which is very heavily dependent upon the particular markers considered and the order in they are considered. A PCA based frame determination also allows marker weighting, if desired, to optimize frame determination based upon user-defined measures of importance, e.g., a marker with a high fidelity is given more weight than a marker with poor fidelity.

The premise of PCA-based frame determination is that the eigenvalues of a matrix characterizing the distribution of markers on the probe are invariant, even though its eigenvectors vary (but remain fixed relative to the marker distribution) according to the pose of the object. We can take advantage of this invariance property to build a local marker frame, defined with the eigenvectors of a matrix that characterizes the marker set, the covariance matrix.

To determine the local coordinate frame of the probe, we assume that the marker locations are known in the global frame. In our case, the global frame coincides with the coordinate frame of the tracking system. Given  $K$  markers on the tracking probe, we denote the markers as  $\mathbf{y}_k$ , with



$i \in [1, K]$ , and  $K \geq 3$ . Each  $\mathbf{y}_k$  is a 3x1 column vector. The centroid of the marker distribution in the global frame is  $\mathbf{y}$ . We also define  $\widetilde{\mathbf{y}}_k$  as the global coordinates of the  $i^{th}$  markers with respect to the centroid. That is,

$$\widetilde{\mathbf{y}}_k = \mathbf{y}_k - \mathbf{y} . \quad (4.6)$$

We refer to the marker location defined with respect to the centroid as markers in a pseudo-local frame.

We start the frame definition process by finding the covariance matrix for the data. Often, a dispersion matrix is used, which is a form of a spatial covariance matrix over the marker distribution. We choose to utilize a dispersion matrix,  $\mathbf{D}$ , defined as

$$\mathbf{D} = \frac{1}{K} \sum_{i=1}^K \widetilde{\mathbf{y}}_k \widetilde{\mathbf{y}}_k^T , \quad (4.7)$$

where  $\mathbf{D}$  is a 3 x 3 symmetrical matrix.

After forming the dispersion matrix, we find its eigenvectors, normalize them, and use them to determine the transformation between the data in the pseudo-local frame and the marker locations fully expressed. Thus, we can express the local coordinates of the probe markers,  $\mathbf{x}_k$ , as

$$\mathbf{x}_k = \mathbf{V}^T \mathbf{y}'_k . \quad (4.8)$$

### 4.1.3 Extension to Probes with Hidden Markers

The frame determination processes described in Sections 4.1.1 and 4.1.2 are valid when the tracking system is able to detect all the markers on a probe simultaneously. However, when tracking systems are subject to occlusion it may not be possible to detect all the markers at once. We now present an extension of the method in Section 4.1.2 to determine a local frame coordinates in the case when some markers may not be detected, or "hidden," in certain probe poses.

First, the tracking probe is placed in an initial pose where at least three markers can be detected by the tracker. We measure the global coordinates of all markers that are able to be detected in this starting pose. Then, an initial local frame is built from the preliminary marker locations using the method from Section 4.1.2. This initial local frame is referred to as a partial local frame. Furthermore, these markers whose global locations are determined from the initial pose of the probe are called "old" markers.

Next, the probe is slowly rotated. This allows previously undetected markers, or "new" markers, the chance to be detected by the tracker. However, the "new" markers must be detected concurrently with at least three "old" markers. When the "new" markers are detected, the rotation matrix,  $\mathbf{R}_{G\_PL}$ , and translation vector,  $\mathbf{t}_{G\_PL}$ , from the partial local frame to the global frame are calculated using the method detailed in Section 4.2.2. A scaling transformation is unnecessary at this stage because we are using normalized coordinate systems.

The inputs to the minimization are the "old" marker coordinates in the partial local frame,  $\mathbf{x}_k$ , and the "old" marker coordinates  $\mathbf{y}_k$  in the global frame measured in the current pose. After the rotation matrix  $\mathbf{R}_{G\_PL}$  and the translation vector  $\mathbf{t}_{G\_PL}$ , are determined, we compute the local coordinates of the "new" markers in the current local frame as

$$\mathbf{x}_k = \mathbf{R}_{G\_PL}^T (\tilde{\mathbf{y}}_k - \mathbf{t}_{G\_PL}) . \quad (4.9)$$

Finally, with all the "new" markers expressed in the local coordinate system, we again compute the dispersion matrix and eigenvalues (Section 4.1.2) to find an estimate of the new local frame and then compute the new local coordinates of the markers. We repeat this process as the real object is slowly rotated until all the markers have been visible. This technique allows us to build a local frame for any kind of marker set and quickly calculate the local coordinates of the markers.

## 4.2 Pose Estimation

Once a local coordinate system has been defined for a tracking probe, the pose of the probe can be determined. The probe pose consists of the location and orientation of the probe, represented as a coordinate transformation from the local frame to the global frame. The pose of a tracking probe may be determined using methods that rely upon direct computation or by methods which use optimization to estimate the best solution. Both approaches are discussed in the following sections.

### 4.2.1 Direct Computation of Pose

By using the data available from three markers, the pose of a tracking probe can be computed directly. Given three markers in the local coordinate frame, denoted as  $\mathbf{x}_k, k \in [1, 3]$ , we can express the marker locations in the global frame,  $\mathbf{y}_k$ , as

$$\mathbf{Y} = \mathbf{R}\mathbf{X} + \mathbf{T} , \quad (4.10)$$

where  $\mathbf{Y}$  is a 3x3 matrix of the global marker locations,  $\mathbf{X}$  is a 3x3 matrix of the local marker locations,  $\mathbf{R}$  is a 3x3 rotation matrix representing the rotation from the local to global frame, and  $\mathbf{T}$  is a 3x3 matrix, with three identical columns composed of the translation from the local frame to the global frame,  $\mathbf{t}$ . If the  $\mathbf{y}_k$  and  $\mathbf{x}_k$  are known, then  $\mathbf{R}$  and  $\mathbf{T}$  can be computed directly by first finding  $\mathbf{R}$  then  $\mathbf{T}$ . Based upon the fact that

$$\begin{aligned} \mathbf{y}_1 &= \mathbf{R}\mathbf{x}_1 + \mathbf{t} \\ \mathbf{y}_2 &= \mathbf{R}\mathbf{x}_2 + \mathbf{t} \\ \mathbf{y}_3 &= \mathbf{R}\mathbf{x}_3 + \mathbf{t} , \end{aligned}$$

we know that there is a  $\mathbf{t}$  such that

$$(\mathbf{y}_k - \mathbf{t}) \cdot (\mathbf{y}_j - \mathbf{t}) = \mathbf{x}_k \cdot \mathbf{x}_j , \quad (4.11)$$

where

$$\mathbf{y}_k - \mathbf{x}_k \neq \mathbf{t} . \quad (4.12)$$

Given this relationship, we can construct two matrices,  $\mathbf{U}$  and  $\mathbf{V}$ . The first two columns of  $\mathbf{U}$  and  $\mathbf{V}$  are expressed as

$$\begin{aligned} \mathbf{u}_1 &= \mathbf{y}_3 - \mathbf{y}_2 & \mathbf{v}_1 &= \mathbf{x}_3 - \mathbf{x}_2 \\ \mathbf{u}_2 &= \mathbf{y}_1 - \mathbf{y}_3 & \mathbf{v}_2 &= \mathbf{x}_1 - \mathbf{x}_3 . \end{aligned}$$

To find the third column of  $\mathbf{U}$  and  $\mathbf{V}$ , we use the fact that

$$\left. \begin{aligned} \mathbf{R}\mathbf{v}_1 &= \mathbf{u}_1 \\ \mathbf{R}\mathbf{v}_2 &= \mathbf{u}_2 \end{aligned} \right\} \Rightarrow \mathbf{R}(\mathbf{v}_1 \times \mathbf{v}_2) = \mathbf{u}_1 \times \mathbf{u}_2 . \quad (4.13)$$

Therefore, we can express the third columns of  $\mathbf{U}$  and  $\mathbf{V}$  as

$$\begin{aligned} \mathbf{u}_3 &= \mathbf{u}_1 \times \mathbf{u}_2 \\ \mathbf{v}_3 &= \mathbf{v}_1 \times \mathbf{v}_2 . \end{aligned}$$

So, we can express  $\mathbf{U}$  and  $\mathbf{V}$  as

$$\begin{aligned} \mathbf{U} &= [\mathbf{u}_1 \ \mathbf{u}_2 \ \mathbf{u}_3] \\ \mathbf{V} &= [\mathbf{v}_1 \ \mathbf{v}_2 \ \mathbf{v}_3] \end{aligned}$$

and we can express  $\mathbf{R}$  as

$$\begin{aligned}\mathbf{R}\mathbf{V} &= \mathbf{U} \\ \mathbf{R} &= \mathbf{U}\mathbf{V}^{-1} .\end{aligned}\tag{4.14}$$

Using this result for  $\mathbf{R}$ , we can then compute  $\mathbf{t}$  as

$$\mathbf{t} = \mathbf{y}_1 - \mathbf{R}\mathbf{x}_1 .\tag{4.15}$$

#### 4.2.2 Optimization Approaches: Least Squares Estimation

The problem with determining pose by direct computation, however, is that the solution is not robust, given that the marker data are noisy in practice. A more robust method for determining the pose of a tracking probe is with optimization. For noisy datasets, optimization that incorporates PCA will minimize the variance between elements of the data. Some popular marker based pose estimation methods that incorporate least-squares optimization are [Spoor and Veldpaus, 1980], [Arun et al., 1987], [Horn, 1987], and [Haralick et al., 1989]. In addition, these methods often use PCA as part of the optimization process.

We use a least-squares optimization method in combination with PCA to find the pose of a tracking probe.  $\mathbf{R}$  and  $\mathbf{t}$  are computed by minimizing the weighted error,  $\epsilon$ , defined as

$$\epsilon(\mathbf{R}, \mathbf{t}) = \sum_{k=1}^K w_k \| \mathbf{y}_k - \mathbf{R}\mathbf{x}_k - \mathbf{t} \|^2 ,\tag{4.16}$$

where  $K$  is the number of markers detected by the tracker,  $\mathbf{y}_k$  is the  $k^{th}$  marker coordinate defined in the global frame, and  $\mathbf{x}_k$  is the  $k^{th}$  marker coordinate defined in the local frame.  $w_k$  is the weight of the  $k^{th}$  marker that quantifies the robustness of the marker data against noise measurements and

its relative motion on the real object. For a rigid probe, we set all the weights to one, although the weights may be varied to provide emphasis to different markers.

We define and compute new coordinates with respect to the weighted centroids,  $x$  and  $y$ , of the markers in each frame given by

$$\widetilde{\mathbf{x}}_k = \mathbf{x}_k - \mathbf{x} \quad (4.17)$$

$$\widetilde{\mathbf{y}}_k = \mathbf{y}_k - \mathbf{y} ,$$

where

$$\mathbf{x} = \frac{1}{W} \sum_{k=1}^K w_k \mathbf{x}_k \quad (4.18)$$

$$\mathbf{y} = \frac{1}{W} \sum_{k=1}^K w_k \mathbf{y}_k \quad (4.19)$$

with

$$W = \sum_{k=1}^K w_k \quad (4.20)$$

The error function (Eq. 4.16) can be rewritten as

$$\epsilon(\mathbf{R}, \mathbf{t}') = \sum_{k=1}^K w_k \|\widetilde{\mathbf{y}}_k - \mathbf{R}\widetilde{\mathbf{x}}_k - \mathbf{t}'\|^2 \quad (4.21)$$

with

$$\mathbf{t}' = \mathbf{t} - \mathbf{y} + \mathbf{R}\mathbf{x} \quad (4.22)$$

We can rewrite the error expression as

$$\epsilon(\mathbf{R}, \mathbf{t}') = \sum_{k=1}^K (\widetilde{\mathbf{y}}_k - \mathbf{R}\widetilde{\mathbf{x}}_k - \mathbf{t}') \cdot (\widetilde{\mathbf{y}}_k - \mathbf{R}\widetilde{\mathbf{x}}_k - \mathbf{t}') . \quad (4.23)$$

Taking the gradient of  $\epsilon$ ,

$$\nabla_{\mathbf{t}'} \epsilon = 2 \sum_{k=1}^K w_k [\widetilde{\mathbf{y}}_k - \mathbf{R} \widetilde{\mathbf{x}}_k - \mathbf{t}'] \quad (4.24)$$

$$= 2 \sum_{k=1}^K w_k \widetilde{\mathbf{y}}_k - 2\mathbf{R} \sum_{k=1}^K w_k \widetilde{\mathbf{x}}_k - 2 \sum_{k=1}^K w_k \mathbf{t}' \quad (4.25)$$

$$= 0 - 0 - 2W\mathbf{t}' \quad (4.26)$$

When the error is minimized,  $\nabla_{\mathbf{t}'} \epsilon = 0$ , which means that  $\mathbf{t}' = 0$ . At this point, we can write  $\epsilon$  as a function of  $\mathbf{R}$ :

$$\epsilon(\mathbf{R}) = \sum_{k=1}^K w_k \|\widetilde{\mathbf{y}}_k - \mathbf{R} \widetilde{\mathbf{x}}_k\|^2 \quad (4.27)$$

Expanding  $\epsilon$ , we get:

$$\epsilon(\mathbf{R}) = \sum_{k=1}^K w_k (\widetilde{\mathbf{y}}_k - \mathbf{R} \widetilde{\mathbf{x}}_k)^T (\widetilde{\mathbf{y}}_k - \mathbf{R} \widetilde{\mathbf{x}}_k) \quad (4.28)$$

$$= \sum_{k=1}^K w_k (\widetilde{\mathbf{y}}_k^T \widetilde{\mathbf{y}}_k + \widetilde{\mathbf{x}}_k^T \mathbf{R}^T \mathbf{R} \widetilde{\mathbf{x}}_k - \widetilde{\mathbf{y}}_k^T \mathbf{R} \widetilde{\mathbf{x}}_k - \widetilde{\mathbf{x}}_k^T \mathbf{R}^T \widetilde{\mathbf{y}}_k) \quad (4.29)$$

which is a scalar. Because  $\mathbf{R}^T \mathbf{R} = I$ , we can rewrite  $\epsilon$  as:

$$\epsilon(\mathbf{R}) = \sum_{k=1}^K w_k (\widetilde{\mathbf{y}}_k^T \widetilde{\mathbf{y}}_k + \widetilde{\mathbf{x}}_k^T \widetilde{\mathbf{x}}_k - 2\widetilde{\mathbf{y}}_k^T \mathbf{R} \widetilde{\mathbf{x}}_k) \quad (4.30)$$

Moreover, because the products  $\widetilde{\mathbf{x}}_k^T \widetilde{\mathbf{x}}_k$  and  $\widetilde{\mathbf{y}}_k^T \widetilde{\mathbf{y}}_k$  are scalars, minimizing  $\epsilon(\mathbf{R})$  is equivalent to maximizing a function of  $\mathbf{R}$  defined as (using Eq. 4.30):

$$f(\mathbf{R}) = \sum_{k=1}^K w_k \widetilde{\mathbf{y}}_k^T \mathbf{R} \widetilde{\mathbf{x}}_k \quad (4.31)$$

At this point, the idea is to solve the maximization of  $f(\mathbf{R})$ , then solve for  $\mathbf{R}$ . Furthermore, when  $\mathbf{R}$  is found,  $\mathbf{t}$  is obtained by

$$\mathbf{t} = \mathbf{y} - \mathbf{R}\mathbf{x} \quad (4.32)$$

$\mathbf{R}$  can be calculated through a process that uses Singular Value Decomposition (SVD). Given a  $n \times m$  matrix,  $\mathbf{A}$ , SVD is a technique by which  $\mathbf{A}$  can be inverted. Formally, SVD allows the row space of a non-invertible matrix to be mapped to a column space. SVD computes the pseudoinverse of  $\mathbf{A}$ , allowing transformations between vector spaces where  $n \neq m$ . In our case, SVD will be used to convert the results of the error functions into a form whose maximum can be easily identified.

Rewriting  $f(\mathbf{R})$  from Equation 4.31:

$$f(\mathbf{R}) = \sum_{k=1}^K w_k \widetilde{\mathbf{y}}_k^T \mathbf{R} \widetilde{\mathbf{x}}_k^T = \text{Trace} \left( \sum_{k=1}^K w_k \mathbf{R} \widetilde{\mathbf{x}}_k \widetilde{\mathbf{y}}_k^T \right) = \text{Trace}(\mathbf{R}\mathbf{H}) \quad , \quad (4.33)$$

where

$$\mathbf{H} = \sum_{k=1}^K w_k \widetilde{\mathbf{x}}_k \widetilde{\mathbf{y}}_k^T \quad . \quad (4.34)$$

It can be shown that  $\text{Trace}(\mathbf{R}\mathbf{H})$  will maximize  $f(\mathbf{R})$  if it is decomposed into a form like  $\mathbf{A}\mathbf{A}^T$ , which is positive definitive [Arun et al., 1987]. Also, by maximizing  $f(\mathbf{R})$ ,  $\mathbf{R}$  will be determined. SVD provides a solution to this problem. We take the SVD of  $\mathbf{H}$  to be

$$\mathbf{H} = \mathbf{U}\mathbf{\Gamma}\mathbf{V}^T \quad , \quad (4.35)$$

where  $\mathbf{U}$  and  $\mathbf{V}$  are orthonormal matrices, and  $\mathbf{\Gamma}$  (the matrix of singular values) is a diagonal matrix with non-negative elements expressed as:

$$\mathbf{\Gamma} = \begin{bmatrix} \gamma_1 & 0 & 0 \\ 0 & \gamma_2 & 0 \\ 0 & 0 & \gamma_3 \end{bmatrix} \quad (4.36)$$



$$= \begin{bmatrix} \sqrt{\gamma_1} & 0 & 0 \\ 0 & \sqrt{\gamma_2} & 0 \\ 0 & 0 & \sqrt{\gamma_3} \end{bmatrix} \begin{bmatrix} \sqrt{\gamma_1} & 0 & 0 \\ 0 & \sqrt{\gamma_2} & 0 \\ 0 & 0 & \sqrt{\gamma_3} \end{bmatrix} \quad (4.37)$$

$$= \mathbf{C}\mathbf{C}^T \quad (4.38)$$

Let  $\mathbf{X} = \mathbf{V}\mathbf{U}^T$ , which is orthonormal. Then,

$$\mathbf{X}\mathbf{H} = \mathbf{V}\mathbf{U}^T\mathbf{U}\mathbf{F}\mathbf{V}^T = \mathbf{V}\mathbf{C}\mathbf{C}^T\mathbf{V}^T . \quad (4.39)$$

Next, let  $\mathbf{A} = \mathbf{V}\mathbf{C}$ . Then,

$$\mathbf{X}\mathbf{H} = \mathbf{A}\mathbf{A}^T . \quad (4.40)$$

$\mathbf{X}\mathbf{H}$  is positive definite and symmetric. Therefore,  $\mathbf{X}$  maximizes  $f(\mathbf{R})$ . As a result,  $\epsilon(\mathbf{R}, \mathbf{t})$  is minimized when  $\mathbf{X} = \mathbf{V}\mathbf{U}^T$ , which yields a solution for  $\mathbf{R}$ . However, there are two possible solutions because  $\mathbf{V}\mathbf{U}^T$  is orthonormal. If  $\det(\mathbf{X}) = 1$ , then  $\mathbf{X} = \mathbf{R}$ . If  $\det(\mathbf{X}) = -1$ , then  $\mathbf{X}$  is a reflection of  $\mathbf{R}$ . We can express a general solution for  $\mathbf{R}$  as

$$\mathbf{R} = \mathbf{V} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(\mathbf{V}\mathbf{U}^T) \end{bmatrix} \mathbf{U}^T . \quad (4.41)$$

The translation,  $\mathbf{t}$ , is then expressed as

$$\mathbf{t} = \mathbf{y} - \mathbf{R}\mathbf{x} . \quad (4.42)$$

### 4.3 Error Analysis in Pose Estimation

After estimating the pose of a tracking probe, methods must be available to assess the accuracy of the results. From the formulation in Section 4.2.2, we are provided with a built-in method for

analyzing the combined effects of errors in position and orientation. Thus the average pose error is computed as

$$\overline{error} = \frac{1}{K} \sum_{k=1}^K w_k \| \mathbf{y}_k - \mathbf{R}\mathbf{x}_k - \mathbf{t} \|^2 . \quad (4.43)$$

The average error has units  $length^2$ , meaning that it expresses the variance of the estimation process for a given probe. This figure of merit by itself has the slight possibility of being misleading, as least-squares measures are subject to interference from outliers. In this case, one could adjust the weight factor attached to the marker whose data is suspect.

In the next chapter, we explore how various constraints in the probe design process and the tracker itself can affect pose estimates. These constraints can be modeled as noise effects and propagated through the pose estimation framework to ascertain their effects upon the estimation. Furthermore, in Chapter 6 we examine how the error analysis is experimentally validated.

## CHAPTER 5: THE EFFECTS OF ENVIRONMENTAL AND APPLICATION FACTORS ON POSE ESTIMATION

In designing conformal tracking probes, we must know how proposed design changes may affect probe performance. This requires defining the factors which affect pose estimation and quantifying the effects of each factor. Considering a static measure of pose for a rigid tracking probe, these factors include the tracker accuracy and precision, and the amount of error in determining the local frame of the probe, the probe topology (that is, the number of markers, how the markers are arranged, and the spatial extent of the tracking probe), and the marker cone of emission, if active markers are used.

When determining the pose of a tracking probe, an ensemble measurement of noisy data is used. Thus, to quantify the effects of the factors, we must first understand how the noise in the data is expressed in the result of the pose estimation process. After quantifying the noise effects, the individual factors for pose estimation can be studied.

We first examine an overall procedure for propagating the effects of noisy data to pose estimation. Using this procedure, we then determine how the tracker accuracy and precision affect the pose estimation process. Next, we study the effects of probe size and shape on pose estimation. We then determine how the number of markers and the marker field of emission affect the pose estimation process. Finally, we summarize the results of this chapter in a tracking factors matrix.

### 5.1 Noise Propagation in Pose Estimation

The data obtained from a tracking system are noisy to varying degrees. The noise will result in errors in determining the global location of markers. Furthermore, this measurement error will propagate to the determination of the local coordinates of markers on a tracking probe. The errors in global and local marker locations will result in an error in the computed pose of the tracking probe. Thus, it is imperative to know how errors in determining marker locations (global and local)

affect the determination of the pose of a probe, which we accomplish by developing a model for error propagation in pose estimation. Previous methods for error propagation do not provide the capability of studying the effects of various factors on the process of pose estimation. Thus we rely upon a customized approach to first-order error propagation.

Our error propagation model is based upon the least-squares minimization procedure to find the pose of a tracking probe. We now summarize the results from Section 4.1.2. Given  $K$  markers, local marker coordinates  $\mathbf{x}_k$ , global marker coordinates  $\mathbf{y}_k$ , and weights  $w_k$  (for  $k = 1, \dots, K$ ), let

$$W = \sum_{k=1}^K w_k \quad (5.1)$$

and

$$\begin{aligned} \mathbf{x} &= \frac{1}{W} \sum_{k=1}^K w_k \mathbf{x}_k \\ \mathbf{y} &= \frac{1}{W} \sum_{k=1}^K w_k \mathbf{y}_k . \end{aligned} \quad (5.2)$$

If we define

$$\widetilde{\mathbf{x}}_k = \mathbf{x}_k - \mathbf{x} \quad (5.3)$$

$$\widetilde{\mathbf{y}}_k = \mathbf{y}_k - \mathbf{y} ,$$

then  $\mathbf{H}$  is given by

$$\mathbf{H} = \sum_{k=1}^K w_k \widetilde{\mathbf{x}}_k \widetilde{\mathbf{y}}_k^T . \quad (5.4)$$

The SVD of  $\mathbf{H}$  is written as

$$\mathbf{H} = \mathbf{U} \mathbf{\Gamma} \mathbf{V}^T . \quad (5.5)$$

The matrices  $\mathbf{U}$ ,  $\mathbf{\Gamma}$ , and  $\mathbf{V}$  are orthogonal (by definition of the SVD) and are determined by the

eigenvalue equations

$$\mathbf{H}^T \mathbf{H} \mathbf{v}_m = \mu_m \mathbf{v}_m \quad (5.6)$$

$$\mathbf{H} \mathbf{H}^T \mathbf{u}_m = \mu_m \mathbf{u}_m$$

for  $m = 1, \dots, M$ , where  $M = 3$ , the dimension of the coordinate vectors in non-homogeneous, cartesian coordinates.

**Proof for Eq. 5.6:**

$$\begin{aligned} \mathbf{H}^T \mathbf{H} &= \mathbf{V} \mathbf{\Gamma}^T \mathbf{U}^T \mathbf{U} \mathbf{\Gamma} \mathbf{V}^T \\ &= \mathbf{V} \mathbf{\Gamma}^T \mathbf{\Gamma} \mathbf{V}^T \\ &= \mathbf{V} (\text{diag}(\mu_1, \mu_2, \mu_3)) \mathbf{V}^T \\ \Rightarrow \mathbf{H}^T \mathbf{H} \mathbf{V} &= \mathbf{V} (\text{diag}(\mu_1, \mu_2, \mu_3)) \\ \Rightarrow \mathbf{H}^T \mathbf{H} \mathbf{v}_m &= \mu_m \mathbf{v}_m \quad \blacksquare \end{aligned} \quad (5.7)$$

where  $\mu_1$ ,  $\mu_2$ , and  $\mu_3$  are the singular values of  $\mathbf{H}$ . With the normalized eigenvectors from the equations in 5.6, we construct  $\mathbf{U}$ ,  $\mathbf{V}$ , and  $\mathbf{\Gamma}$

$$\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_m] \quad (5.8)$$

$$\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_m]$$

$$[\mathbf{\Gamma}]_{mn} = \mu_m \delta_{mn}$$

for  $n = 1, \dots, M$ . The singular vectors can always be chosen to satisfy

$$\mathbf{v}_m^T \mathbf{v}_n = \delta_{mn} \quad (5.9)$$

$$\mathbf{u}_m^T \mathbf{u}_n = \delta_{mn} \quad ,$$

which implies that  $\mathbf{U}$  and  $\mathbf{V}$  are unitary, meaning

$$\begin{aligned}\mathbf{U}^T\mathbf{U} &= \mathbf{U}\mathbf{U}^T = \mathbf{U}\mathbf{U}^{-1} = \mathbf{U}\mathbf{U}^\dagger = I \\ \mathbf{V}^T\mathbf{V} &= \mathbf{V}\mathbf{V}^T = \mathbf{V}\mathbf{V}^{-1} = \mathbf{V}\mathbf{V}^\dagger = I ,\end{aligned}\tag{5.10}$$

where  $\mathbf{U}^\dagger$  and  $\mathbf{V}^\dagger$  indicate the complex conjugate transpose of  $\mathbf{U}$  and  $\mathbf{V}$ . From  $\mathbf{U}$  and  $\mathbf{V}$ , we can compute  $\mathbf{R}$  as

$$\mathbf{R} = \mathbf{V}\mathbf{D}\mathbf{U}^T ,\tag{5.11}$$

where

$$\mathbf{D} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(\mathbf{V}\mathbf{U}^T) \end{bmatrix}\tag{5.12}$$

and  $\mathbf{t} = \mathbf{y} - \mathbf{R}\mathbf{x}$ .

To propagate the error, we first apply the local marker coordinate error,  $\Delta\mathbf{x}_k$ , and global marker coordinate error,  $\Delta\mathbf{y}_k$ , to each marker as

$$\mathbf{x}_k \rightarrow \mathbf{x}_k + \Delta\mathbf{x}_k\tag{5.13}$$

$$\mathbf{y}_k \rightarrow \mathbf{y}_k + \Delta\mathbf{y}_k ,\tag{5.14}$$

assuming that  $\mathbf{x}_k \gg \Delta\mathbf{x}_k$  and  $\mathbf{y}_k \gg \Delta\mathbf{y}_k$ . We also apply the local and global coordinate errors with respect to the marker centroids,  $\widetilde{\mathbf{x}}_k$  and  $\widetilde{\mathbf{y}}_k$  as

$$\widetilde{\mathbf{x}}_k \rightarrow \widetilde{\mathbf{x}}_k + \Delta\widetilde{\mathbf{x}}_k\tag{5.15}$$

$$\widetilde{\mathbf{y}}_k \rightarrow \widetilde{\mathbf{y}}_k + \Delta\widetilde{\mathbf{y}}_k\tag{5.16}$$

where, similar to the relationship in Eq. 5.3,

$$\Delta \widetilde{\mathbf{x}}_k = \Delta \mathbf{x}_k - \frac{1}{W} \sum_{k=1}^K w_k \Delta \mathbf{x}_k \quad (5.17)$$

$$\Delta \widetilde{\mathbf{y}}_k = \Delta \mathbf{y}_k - \frac{1}{W} \sum_{k=1}^K w_k \Delta \mathbf{y}_k . \quad (5.18)$$

$w_k$  is the weight of the  $k^{th}$  marker that quantifies the robustness of the marker data against noise measurements and its relative motion on the real object. For a rigid probe, all weights are set to 1.

Because we wish to determine how the errors propagate to the rotation and translation of the tracking probe,  $\mathbf{R}$  and  $\mathbf{t}$ , we must determine how the errors affect the matrix  $\mathbf{H}$ . Thus,

$$\mathbf{H} \rightarrow \mathbf{H} + \Delta \mathbf{H} \quad (5.19)$$

If we propagate the errors in  $\mathbf{x}$  and  $\mathbf{y}$  to  $\mathbf{H}$  we get

$$\mathbf{H} + \Delta \mathbf{H} = \sum_{k=1}^K w_k (\widetilde{\mathbf{x}}_k + \Delta \widetilde{\mathbf{x}}_k) (\widetilde{\mathbf{y}}_k + \Delta \widetilde{\mathbf{y}}_k)^T \quad (5.20)$$

$$= \sum_{k=1}^K w_k (\widetilde{\mathbf{x}}_k \widetilde{\mathbf{y}}_k^T + \Delta \widetilde{\mathbf{x}}_k \widetilde{\mathbf{y}}_k^T + \widetilde{\mathbf{x}}_k \Delta \widetilde{\mathbf{y}}_k^T + \Delta \widetilde{\mathbf{x}}_k \Delta \widetilde{\mathbf{y}}_k^T) \quad (5.21)$$

By first order approximation, we obtain

$$\mathbf{H} + \Delta \mathbf{H} \approx \sum_{k=1}^K w_k (\widetilde{\mathbf{x}}_k \widetilde{\mathbf{y}}_k^T + \Delta \widetilde{\mathbf{x}}_k \widetilde{\mathbf{y}}_k^T + \widetilde{\mathbf{x}}_k \Delta \widetilde{\mathbf{y}}_k^T) \quad (5.22)$$

$$\approx \sum_{k=1}^K w_k (\Delta \widetilde{\mathbf{x}}_k \widetilde{\mathbf{y}}_k^T + \widetilde{\mathbf{x}}_k \Delta \widetilde{\mathbf{y}}_k^T) + \mathbf{H} \quad (5.23)$$

which defines  $\Delta \mathbf{H}$  as

$$\Delta \mathbf{H} \approx \sum_{k=1}^K w_k (\Delta \widetilde{\mathbf{x}}_k \widetilde{\mathbf{y}}_k^T + \widetilde{\mathbf{x}}_k \Delta \widetilde{\mathbf{y}}_k^T) . \quad (5.24)$$

This is a first order model of error propagation. For the rest of this section, an equation of the form  $a \approx b$  is taken to mean that the two quantities  $a$  and  $b$  are equal to the first order in the measurement errors.

We now define a matrix  $\mathbf{Q}$  as

$$\mathbf{Q} = \mathbf{H}^T \mathbf{H} . \quad (5.25)$$

Then, with first order error propagation,

$$\mathbf{Q} \rightarrow \mathbf{Q} + \Delta \mathbf{Q} \quad (5.26)$$

with

$$\Delta \mathbf{Q} \approx \Delta \mathbf{H}^T \mathbf{H} + \mathbf{H}^T \Delta \mathbf{H} \quad (5.27)$$

which is a first order expansion of  $\Delta \mathbf{Q}$ . Now, consider the eigenvector equations from Eq. 5.6. We propagate the error as

$$(\mathbf{Q} + \Delta \mathbf{Q})(\mathbf{v}_m + \Delta \mathbf{v}_m) = (\mu_m + \Delta \mu_m)(\mathbf{v}_m + \Delta \mathbf{v}_m) \quad (5.28)$$

$$(\mathbf{Q}^T + \Delta \mathbf{Q}^T)(\mathbf{u}_m + \Delta \mathbf{u}_m) = (\mu_m + \Delta \mu_m)(\mathbf{u}_m + \Delta \mathbf{u}_m) , \quad (5.29)$$

which must be satisfied with the constraints

$$(\mathbf{v}_m + \Delta \mathbf{v}_m)^T (\mathbf{v}_n + \Delta \mathbf{v}_n) = \delta_{mn} \quad (5.30)$$

$$(\mathbf{u}_m + \Delta \mathbf{u}_m)^T (\mathbf{u}_n + \Delta \mathbf{u}_n) = \delta_{mn}$$

because orthogonality must still hold, even with errors in the  $\mathbf{H}$  matrix. Using the error-free



eigenvector equations (Eqs. 5.6 and 5.9), and applying the previous constraints, Eqs. 5.28 and 5.29 can be expanded in the following manner

$$\mathbf{Q}\mathbf{v}_m + \Delta\mathbf{Q}\mathbf{v}_m + \mathbf{Q}\Delta\mathbf{v}_m + \Delta\mathbf{Q}\Delta\mathbf{v}_m = \mu_m\mathbf{v}_m + \Delta\mu_m\mathbf{v}_m + \mu_m\Delta\mathbf{v}_m + \Delta\mu_m\Delta\mathbf{v}_m \quad (5.31)$$

$$\mathbf{Q}^T\mathbf{u}_m + \Delta\mathbf{Q}^T\mathbf{u}_m + \mathbf{Q}^T\Delta\mathbf{u}_m + \Delta\mathbf{Q}^T\Delta\mathbf{u}_m = \mu_m\mathbf{u}_m + \Delta\mu_m\mathbf{u}_m + \mu_m\Delta\mathbf{u}_m + \Delta\mu_m\Delta\mathbf{u}_m \quad (5.32)$$

Because

$$\mathbf{Q}\mathbf{v}_m = \mu_m\mathbf{v}_m \quad (5.33)$$

$$\mathbf{Q}^T\mathbf{u}_m = \mu_m\mathbf{u}_m$$

$$\Delta\mathbf{Q}\Delta\mathbf{v}_m \approx \Delta\mathbf{Q}^T\Delta\mathbf{u}_m \approx 0 \text{ and}$$

$$\Delta\mu_m\Delta\mathbf{v}_m \approx \Delta\mu_m\Delta\mathbf{u}_m \approx 0 ,$$

we can simplify Eq. 5.31 to

$$\mu_m\mathbf{v}_m + \Delta\mathbf{Q}\mathbf{v}_m + \mathbf{Q}\Delta\mathbf{v}_m \approx \mu_m\mathbf{v}_m + \Delta\mu_m\mathbf{v}_m + \mu_m\Delta\mathbf{v}_m \quad (5.34)$$

$$\Rightarrow \Delta\mathbf{Q}\mathbf{v}_m + \mathbf{Q}\Delta\mathbf{v}_m \approx \Delta\mu_m\mathbf{v}_m + \mu_m\Delta\mathbf{v}_m$$

and, likewise, Eq. 5.32 to

$$\mu_m\mathbf{u}_m + \Delta\mathbf{Q}^T\mathbf{u}_m + \mathbf{Q}^T\Delta\mathbf{u}_m \approx \mu_m\mathbf{u}_m + \Delta\mu_m\mathbf{u}_m + \mu_m\Delta\mathbf{u}_m \quad (5.35)$$

$$\Rightarrow \Delta\mathbf{Q}^T\mathbf{u}_m + \mathbf{Q}^T\Delta\mathbf{u}_m \approx \Delta\mu_m\mathbf{u}_m + \mu_m\Delta\mathbf{u}_m$$

with the orthogonality constraints (derived by first order expansion of Eq. 5.9)

$$\mathbf{v}_m^T\Delta\mathbf{v}_n + \mathbf{v}_n^T\Delta\mathbf{v}_m \approx 0 \quad (5.36)$$

$$\mathbf{u}_m^T\Delta\mathbf{u}_n + \mathbf{u}_n^T\Delta\mathbf{u}_m \approx 0 .$$

Our goal is to solve this system of equations for  $\Delta\mu_m$  and  $\Delta\mathbf{v}_m$  (the first order error of the eigenvalue and its associated eigenvector) in terms of  $\Delta\mathbf{Q}$  (the first order error in the matrix describing the relation between the two coordinate systems). To accomplish this task, we define the matrices  $\mathbf{A}$  and  $\mathbf{B}$  by the equations

$$\Delta\mathbf{v}_m = \mathbf{A}\mathbf{v}_m, \quad m = 1, \dots, M \quad (5.37)$$

$$\Delta\mathbf{u}_m = \mathbf{B}\mathbf{u}_m, \quad m = 1, \dots, M$$

These equations define  $\mathbf{A}$  and  $\mathbf{B}$  uniquely since the singular vectors form basis  $\mathbb{R}^M$ . By substituting into Eqs. 5.34 and 5.35 with the previous equation, the system can be written in terms of these new matrices as

$$\Delta\mathbf{Q}\mathbf{v}_m + \mathbf{Q}\mathbf{A}\mathbf{v}_m \approx \Delta\mu_m\mathbf{v}_m + \mu_m\mathbf{A}\mathbf{v}_m \quad (5.38)$$

$$\Delta\mathbf{Q}^T\mathbf{u}_m + \mathbf{Q}^T\mathbf{B}\mathbf{u}_m \approx \Delta\mu_m\mathbf{u}_m + \mu_m\mathbf{B}\mathbf{u}_m \quad (5.39)$$

and the orthogonality constraints become

$$\mathbf{v}_m^T\mathbf{A}\mathbf{v}_n + \mathbf{v}_n^T\mathbf{A}\mathbf{v}_m \approx 0 \quad (5.40)$$

$$\mathbf{u}_m^T\mathbf{B}\mathbf{u}_n + \mathbf{u}_n^T\mathbf{B}\mathbf{u}_m \approx 0$$

By rearranging Eqs. 5.38 and 5.39, we have

$$(\mu_m - \mathbf{Q})\mathbf{A}\mathbf{v}_m \approx (\Delta\mathbf{Q} - \Delta\mu_m)\mathbf{v}_m \quad (5.41)$$

$$(\mu_m - \mathbf{Q}^T)\mathbf{B}\mathbf{u}_m \approx (\Delta\mathbf{Q}^T - \Delta\mu_m)\mathbf{u}_m \quad (5.42)$$

We now take Eq. 5.41 and multiply through by  $\mathbf{v}_n^T$ ,

$$\mathbf{v}_n^T\Delta\mathbf{Q}\mathbf{v}_m - \mathbf{v}_n^T\Delta\mu_m\mathbf{v}_m \approx \mathbf{v}_n^T\mu_m\mathbf{A}\mathbf{v}_m - \mathbf{v}_n^T\mathbf{Q}\mathbf{A}\mathbf{v}_m \quad (5.43)$$

Since

$$\mathbf{Q}\mathbf{v}_n = \mu_n\mathbf{v}_n \Rightarrow (\mathbf{Q}\mathbf{v}_n)^T = (\mu_n\mathbf{v}_n)^T \Rightarrow \mathbf{v}_n^T\mathbf{Q}^T = \mu_n\mathbf{v}_n^T, \quad (5.44)$$

we can express Eq. 5.43 as

$$\mathbf{v}_n^T\Delta\mathbf{Q}\mathbf{v}_m - \Delta\mu_m\mathbf{v}_n^T\mathbf{v}_m \approx \mu_m\mathbf{v}_n^T\mathbf{A}\mathbf{v}_m - \mu_n\mathbf{v}_n^T\mathbf{A}\mathbf{v}_m. \quad (5.45)$$

Furthermore, because  $\mathbf{v}_n^T\mathbf{v}_m = \delta_{mn}$ ,

$$\mathbf{v}_n^T\Delta\mathbf{Q}\mathbf{v}_m - \Delta\mu_m\delta_{mn} \approx (\mu_m - \mu_n)\mathbf{v}_n^T\mathbf{A}\mathbf{v}_m. \quad (5.46)$$

When  $m = n$ , we get

$$\Delta\mu_m \approx \mathbf{v}_n^T\Delta\mathbf{Q}\mathbf{v}_m. \quad (5.47)$$

When  $m \neq n$ , we get

$$\mathbf{v}_n^T\mathbf{A}\mathbf{v}_m \approx \frac{\mathbf{v}_n^T\Delta\mathbf{Q}\mathbf{v}_m}{\mu_m - \mu_n}, \quad (5.48)$$

which fully specifies  $\mathbf{A}$ . Note that since  $\Delta\mathbf{Q}^T = \Delta\mathbf{Q}$  (meaning  $\mathbf{Q}$  is symmetric), we have

$$\mathbf{v}_m^T\Delta\mathbf{Q}\mathbf{v}_n = \mathbf{v}_m^T\Delta\mathbf{Q}^T\mathbf{v}_n = (\mathbf{v}_n^T\Delta\mathbf{Q}\mathbf{v}_m)^T. \quad (5.49)$$

Moreover, because  $\mathbf{v}_m^T\Delta\mathbf{Q}\mathbf{v}_n$  is a scalar,

$$(\mathbf{v}_n^T\Delta\mathbf{Q}\mathbf{v}_m)^T = \mathbf{v}_n^T\Delta\mathbf{Q}\mathbf{v}_m. \quad (5.50)$$

As a result, the orthogonality constraints on  $\mathbf{A}$  are satisfied. Now, we collect the terms of  $\Delta\mathbf{H}$ .

Starting with the expression of  $\mathbf{H}$ , we see that

$$\mathbf{H} = \mathbf{U}\mathbf{F}\mathbf{V}^T, \quad (5.51)$$

implying that

$$\begin{aligned}\mathbf{H}\mathbf{V} &= \mathbf{U}\mathbf{\Gamma} \\ \Rightarrow \mathbf{U}^T\mathbf{H} &= \mathbf{\Gamma}\mathbf{V}^T \\ \Rightarrow \mathbf{H}^T\mathbf{U} &= \mathbf{V}\mathbf{\Gamma}\end{aligned}$$

and leading to

$$\begin{aligned}\mathbf{H}\mathbf{v}_m &= \mathbf{u}_m\sqrt{\mu_m} = \sqrt{\mu_m}\mathbf{u}_m \\ \mathbf{H}^T\mathbf{u}_m &= \mathbf{v}_m\sqrt{\mu_m} = \sqrt{\mu_m}\mathbf{v}_m .\end{aligned}\tag{5.52}$$

We can now develop two scenarios. The first is when  $m = n$ . We start with the relation from Eq. 5.47, expanding with the relationships from Eqs. 5.27 and 5.52

$$\Delta\mu_m \approx \mathbf{v}_n^T\Delta\mathbf{Q}\mathbf{v}_m \approx \mathbf{v}_m^T\Delta\mathbf{Q}\mathbf{v}_m , m = n\tag{5.53}$$

$$\approx \mathbf{v}_m^T(\Delta\mathbf{H}^T\mathbf{H} + \mathbf{H}^T\Delta\mathbf{H})\mathbf{v}_m\tag{5.54}$$

$$\approx \mathbf{v}_m^T\Delta\mathbf{H}^T\mathbf{H}\mathbf{v}_m + \mathbf{v}_m^T\mathbf{H}^T\Delta\mathbf{H}\mathbf{v}_m\tag{5.55}$$

$$\approx \mathbf{v}_m^T\Delta\mathbf{H}^T\sqrt{\mu_m}\mathbf{u}_m + \sqrt{\mu_m}\mathbf{u}_m^T\Delta\mathbf{H}\mathbf{v}_m\tag{5.56}$$

$$\approx \sqrt{\mu_m}(\mathbf{v}_m^T\Delta\mathbf{H}^T\mathbf{u}_m + \mathbf{u}_m^T\Delta\mathbf{H}\mathbf{v}_m)\tag{5.57}$$

$$\approx 2\sqrt{\mu_m}\mathbf{u}_m^T\Delta\mathbf{H}\mathbf{v}_m .\tag{5.58}$$

The second scenario is when  $m \neq n$ . Starting with the relation from Eq. 5.48, expanding with the relationships from Eqs. 5.27 and 5.52

$$\mathbf{v}_n^T\mathbf{A}\mathbf{v}_m \approx \frac{\mathbf{v}_n^T\Delta\mathbf{Q}\mathbf{v}_m}{\mu_m - \mu_n}\tag{5.59}$$

$$\approx \frac{\mathbf{v}_n^T(\Delta\mathbf{H}^T\mathbf{H} + \mathbf{H}^T\Delta\mathbf{H})\mathbf{v}_m}{\mu_m - \mu_n}\tag{5.60}$$

$$\approx \frac{\mathbf{v}_n^T \Delta \mathbf{H}^T \mathbf{H} \mathbf{v}_m + \mathbf{v}_n^T \mathbf{H}^T \Delta \mathbf{H} \mathbf{v}_m}{\mu_m - \mu_n} \quad (5.61)$$

$$\approx \frac{\mathbf{v}_n^T \Delta \mathbf{H}^T \sqrt{\mu_m} \mathbf{u}_m + \sqrt{\mu_n} \mathbf{u}_n^T \Delta \mathbf{H} \mathbf{v}_m}{\mu_m - \mu_n} \quad (5.62)$$

$$\approx \frac{\sqrt{\mu_n} \mathbf{u}_n^T \Delta \mathbf{H} \mathbf{v}_m + \sqrt{\mu_m} \mathbf{u}_m^T \Delta \mathbf{H} \mathbf{v}_n}{\mu_m - \mu_n} . \quad (5.63)$$

By replacing  $\mathbf{H}$  with  $\mathbf{H}^T$  and interchanging  $\mathbf{v}_k$  with  $\mathbf{u}_k$  in Eq. 5.63, we get the following equation for  $\mathbf{B}$

$$\mathbf{v}_n^T \mathbf{B} \mathbf{v}_m \approx \frac{\sqrt{\mu_m} \mathbf{u}_n^T \Delta \mathbf{H} \mathbf{v}_m + \sqrt{\mu_n} \mathbf{u}_m^T \Delta \mathbf{H} \mathbf{v}_n}{\mu_m - \mu_n} . \quad (5.64)$$

Finally, if

$$\mathbf{V} \rightarrow \mathbf{V} + \Delta \mathbf{V} \quad (5.65)$$

$$\mathbf{U} \rightarrow \mathbf{U} + \Delta \mathbf{U}$$

then

$$\Delta \mathbf{V} \approx \mathbf{A} \mathbf{V} \quad (5.66)$$

$$\Delta \mathbf{U} \approx \mathbf{B} \mathbf{U} .$$

At this point, we now have a direct relationship between the eigenvectors of  $\mathbf{V}$  and  $\mathbf{U}$  and the first order error propagated to the eigenvectors by way of  $\mathbf{A}$  and  $\mathbf{B}$ .  $\mathbf{A}$  and  $\mathbf{B}$  are also anti-symmetric, meaning  $\mathbf{A} = -\mathbf{A}^T$ .

From the expression of  $\mathbf{R}$  in Eq. 5.11, we can derive an expression for the propagation of the first order error to the calculation of  $\mathbf{R}$ . To start, we can express the rotation matrix with error,  $\mathbf{R}_{err}$  as a concatenation of an incremental rotation,  $\Delta \mathbf{R}$ , and an "error free" rotation,  $\mathbf{R}$

$$\mathbf{R}_{err} = \Delta \mathbf{R} \mathbf{R} . \quad (5.67)$$

Using Eq. 5.11, we can apply first order errors to  $\mathbf{V}$  and  $\mathbf{U}$  and obtain the following expression for  $\mathbf{R}_{err}$ :

$$\mathbf{R}_{err} = \Delta\mathbf{R}\mathbf{R} = (\mathbf{V} + \Delta\mathbf{V})\mathbf{D}(\mathbf{U} + \Delta\mathbf{U})^T \quad (5.68)$$

$$= (\mathbf{V} + \Delta\mathbf{V})(\mathbf{D}\mathbf{U}^T + \mathbf{D}\Delta\mathbf{U}^T) \quad (5.69)$$

$$= \mathbf{V}\mathbf{D}\mathbf{U}^T + \Delta\mathbf{V}\mathbf{D}\mathbf{U}^T + \mathbf{V}\mathbf{D}\Delta\mathbf{U}^T + \Delta\mathbf{V}\mathbf{D}\Delta\mathbf{U}^T \quad (5.70)$$

$$= \mathbf{V}\mathbf{D}\mathbf{U}^T + \mathbf{A}\mathbf{V}\mathbf{D}\mathbf{U}^T + \mathbf{V}\mathbf{D}\mathbf{U}^T\mathbf{B}^T + \mathbf{A}\mathbf{V}\mathbf{D}\mathbf{U}^T\mathbf{B}^T \quad (5.71)$$

$$= \mathbf{V}\mathbf{D}\mathbf{U}^T + \mathbf{V}\mathbf{D}\mathbf{U}^T\mathbf{B}^T + \mathbf{A}\mathbf{V}\mathbf{D}\mathbf{U}^T + \mathbf{A}\mathbf{V}\mathbf{D}\mathbf{U}^T\mathbf{B}^T \quad (5.72)$$

$$= \mathbf{V}\mathbf{D}\mathbf{U}^T - \mathbf{V}\mathbf{D}\mathbf{U}^T\mathbf{B} + \mathbf{A}\mathbf{V}\mathbf{D}\mathbf{U}^T - \mathbf{A}\mathbf{V}\mathbf{D}\mathbf{U}^T\mathbf{B} \quad (5.73)$$

$$= \mathbf{V}\mathbf{D}\mathbf{U}^T(\mathbf{I} - \mathbf{B}) + \mathbf{A}\mathbf{V}\mathbf{D}\mathbf{U}^T(\mathbf{I} - \mathbf{B}) \quad (5.74)$$

$$= (\mathbf{V}\mathbf{D}\mathbf{U}^T + \mathbf{A}\mathbf{V}\mathbf{D}\mathbf{U}^T)(\mathbf{I} - \mathbf{B}) \quad (5.75)$$

$$= (\mathbf{I} + \mathbf{A})\mathbf{V}\mathbf{D}\mathbf{U}^T(\mathbf{I} - \mathbf{B}) \quad (5.76)$$

$$= (\mathbf{I} + \mathbf{A})\mathbf{R}(\mathbf{I} - \mathbf{B}) . \quad (5.77)$$

Taking into account the fact that a matrix exponential can be approximated by a Taylor series expansion, a first order approximation of  $\mathbf{R}_{err}$  is ,

$$\mathbf{R}_{err} \approx e^{\mathbf{A}}\mathbf{R}e^{-\mathbf{B}} \quad (5.78)$$

$$\approx e^{\mathbf{A}}\mathbf{R}e^{-\mathbf{B}}\mathbf{I} \quad (5.79)$$

$$\approx e^{\mathbf{A}}\mathbf{R}e^{-\mathbf{B}}\mathbf{R}^T\mathbf{R} \quad (5.80)$$

$$\approx \underbrace{e^{\mathbf{A}}\mathbf{R}e^{-\mathbf{B}}\mathbf{R}^T}_{\Delta\mathbf{R}}\mathbf{R} \quad (5.81)$$

Because  $\mathbf{A}$  and  $\mathbf{B}$  are anti-symmetric matrices,  $e^{\mathbf{A}}$  and  $e^{-\mathbf{B}}$  yield valid rotation matrices in most cases (invalid cases occur when  $\Delta\mathbf{x}$  or  $\Delta\mathbf{y}$  are greater than or equal to the order of magnitude of

$\tilde{\mathbf{x}}$  and  $\tilde{\mathbf{y}}$ ). Thus,  $\mathbf{R}_{err}$  is a valid rotation matrix that expresses the orientation of the probe with marker location errors propagated through its solution.

The position of the probe without error,  $\mathbf{t}$ , is expressed as

$$\mathbf{t} = \mathbf{y} - \mathbf{R}\mathbf{x} , \quad (5.82)$$

where  $\mathbf{y}$  and  $\mathbf{x}$  are the centroids of the  $\mathbf{y}_k$  and  $\mathbf{x}_k$ , respectively. If we propagate the first order errors to the probe position, we obtain  $\mathbf{t}_{err}$ , which can be written as

$$\begin{aligned} \mathbf{t}_{err} &= \frac{1}{W} \sum_{k=1}^K w_k (\mathbf{y}_k + \Delta \mathbf{y}_k) - \mathbf{R}_{err} \frac{1}{W} \sum_{k=1}^K w_k (\mathbf{x}_k + \Delta \mathbf{x}_k) \\ &= \frac{1}{W} \sum_{k=1}^K w_k \mathbf{y}_k + \frac{1}{W} \sum_{k=1}^K w_k \Delta \mathbf{y}_k - \mathbf{R}_{err} \frac{1}{W} \sum_{k=1}^K w_k \mathbf{x}_k - \mathbf{R}_{err} \frac{1}{W} \sum_{k=1}^K w_k \Delta \mathbf{x}_k \\ &= \mathbf{y} + \frac{1}{W} \sum_{k=1}^K w_k \Delta \mathbf{y}_k - \mathbf{R}_{err} \mathbf{x} - \mathbf{R}_{err} \frac{1}{W} \sum_{k=1}^K w_k \Delta \mathbf{x}_k \\ &= \mathbf{y} + \Delta \mathbf{y}_c - \mathbf{R}_{err} \mathbf{x} - \mathbf{R}_{err} \Delta \mathbf{x}_c \\ &= \mathbf{y} + \Delta \mathbf{y}_c - \mathbf{R}_{err} (\mathbf{x} + \Delta \mathbf{x}_c) , \end{aligned} \quad (5.83)$$

where  $\Delta \mathbf{x}_c$  and  $\Delta \mathbf{y}_c$  are the centroids of the  $\Delta \mathbf{x}_k$  and  $\Delta \mathbf{y}_k$ , respectively.

## 5.2 Effects of Factors on Pose Estimation

In analyzing the effects of environmental factors on pose estimation, we must examine the properties of the  $\mathbf{R}_{err}$  matrix. As a result, our analysis focuses on  $\mathbf{A}$  and  $\mathbf{B}$ , as these matrices determine  $\mathbf{R}_{err}$ .

If the noise added to the marker locations,  $\Delta \mathbf{x}_k$  and  $\Delta \mathbf{y}_k$ , are zero, then  $\mathbf{A}$  and  $\mathbf{B}$  become zero matrices, giving  $e^{\mathbf{A}} = e^{-\mathbf{B}} = \mathbf{I}$  and  $\mathbf{R}_{err} = \mathbf{R}$ , indicating zero pose error. Furthermore, the zero matrices will have three eigenvalues at zero. If the magnitude of  $\Delta \mathbf{x}$  and  $\Delta \mathbf{y}$  become infinite, the magnitude of the two non-zero eigenvalues of  $\mathbf{A}$  and  $\mathbf{B}$  are infinite. When  $\Delta \mathbf{x}$  and  $\Delta \mathbf{y}$

are between zero and infinity,  $\mathbf{A}$  and  $\mathbf{B}$  give valid rotation matrices with measurable eigenvalues. As the magnitude of the elements of  $\mathbf{A}$  and  $\mathbf{B}$  increase or decrease, the magnitude of the non-zero eigenvalues of  $\mathbf{A}$  and  $\mathbf{B}$  increase or decrease, respectively. Thus, by examining the effect of various factors on the eigenvalues of  $\mathbf{A}$  and  $\mathbf{B}$ , we can determine the effects on the overall pose error.

We now define the eigenvalues of  $\mathbf{A}$ . Starting with Equation 5.48, we know that

$$\mathbf{v}_n^T \mathbf{A} \mathbf{v}_m \approx \frac{\mathbf{v}_n^T \Delta \mathbf{Q} \mathbf{v}_m}{\mu_m - \mu_n}, n \neq m. \quad (5.84)$$

Therefore,

$$\mathbf{v}_n^T \mathbf{A} \mathbf{v}_m \approx \frac{\mathbf{v}_n^T (\Delta \mathbf{H}^T \mathbf{H} + \mathbf{H}^T \Delta \mathbf{H}) \mathbf{v}_m}{\mu_m - \mu_n}, n \neq m. \quad (5.85)$$

Expressing each matrix generally, the  $\mathbf{H}$  matrix is

$$\mathbf{H} = \begin{bmatrix} s_{11} & s_{12} & s_{13} \\ s_{21} & s_{22} & s_{23} \\ s_{31} & s_{32} & s_{33} \end{bmatrix} \quad (5.86)$$

with

$$s_{nm} = \sum_{k=1}^K w_k \widetilde{x}_{kn} \widetilde{y}_{km}, \quad (5.87)$$

where  $\widetilde{x}_{kn}$  and  $\widetilde{y}_{km}$  are the  $n^{th}$  and  $m^{th}$  elements of  $\widetilde{\mathbf{x}}_k$  and  $\widetilde{\mathbf{y}}_k$ , respectively. The  $\Delta \mathbf{H}$  matrix can be written as

$$\Delta \mathbf{H} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix} \quad (5.88)$$

with

$$\sigma_{nm} \approx \sum_{k=1}^K w_k (\Delta \widetilde{x}_{kn} \widetilde{y}_{km} + \widetilde{x}_{kn} \Delta \widetilde{y}_{km}), \quad (5.89)$$



where  $n$  and  $m$  subscripts represent the  $n^{th}$  and  $m^{th}$  elements of each vector, respectively. The matrix  $\mathbf{A}$  can be generally expressed as

$$\mathbf{A} = \begin{bmatrix} 0 & a & b \\ -a & 0 & c \\ -b & -c & 0 \end{bmatrix} \quad (5.90)$$

in the  $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$  basis. Then,

$$a = \frac{\mathbf{v}_1^T \Delta \mathbf{Q} \mathbf{v}_2}{\mu_2 - \mu_1} \quad (5.91)$$

$$b = \frac{\mathbf{v}_1^T \Delta \mathbf{Q} \mathbf{v}_3}{\mu_3 - \mu_1} \quad (5.92)$$

$$c = \frac{\mathbf{v}_2^T \Delta \mathbf{Q} \mathbf{v}_3}{\mu_3 - \mu_2} . \quad (5.93)$$

Finally, the eigenvalues of  $\mathbf{A}$  are

$$\lambda_1 = \sqrt{-a^2 - b^2 - c^2} = \sqrt{-(a^2 + b^2 + c^2)} = j\sqrt{a^2 + b^2 + c^2} \quad (5.94)$$

$$\lambda_2 = -\sqrt{-a^2 - b^2 - c^2} = -j\sqrt{a^2 + b^2 + c^2}$$

$$\lambda_3 = 0 .$$

A similar formulation applies to finding the eigenvalues for  $\mathbf{B}$ , with the only difference being that  $\Delta \mathbf{Q}^T$  is used instead of  $\Delta \mathbf{Q}$ .

### 5.2.1 The Effect of Tracker Accuracy and Frame Determination on Pose Estimation

To determine the effect of tracker accuracy on pose estimation, we must determine the relationship between  $\Delta \mathbf{y}$  and the eigenvalues of  $\mathbf{A}$  and  $\mathbf{B}$ . Specifically, the change in the magnitude of the eigenvalues, whether increasing or decreasing, is of interest.

We now examine the eigenvalues of  $\mathbf{A}$  and how they are affected by the tracker noise, given that a similar formulation exists for  $\mathbf{B}$ . As expressed in Equation 5.94, the magnitude of the non-zero eigenvalues of  $\mathbf{A}$  increase or decrease as the squared magnitude increases or decreases. In Equation 5.89 we can see that the influence of  $\Delta\mathbf{y}$  is on the expression for  $\sigma_{nm}$ . When analyzing the effect on the eigenvalues, we consider all other quantities constant, allowing for variation in  $\sigma_{nm}$  only. We consider  $\Delta\mathbf{y}$  a Gaussian random variable with zero mean and a standard deviation equal to the RMS tracker accuracy. Given these conditions, we can express the squared magnitude of the non-zero eigenvectors as

$$|\lambda_{1,2}|^2 = a^2 + b^2 + c^2 . \quad (5.95)$$

Because  $\Delta\mathbf{y}$  is a random variable, we must treat eigenvalues as an ensemble average of event occurrences. Thus, the previous equation becomes

$$\langle |\lambda_{1,2}|^2 \rangle = \langle a^2 + b^2 + c^2 \rangle = \langle a^2 \rangle + \langle b^2 \rangle + \langle c^2 \rangle . \quad (5.96)$$

Given

$$\alpha_{mn} = \frac{1}{\mu_m - \mu_n} , \quad (5.97)$$

We can express  $a^2$  as

$$\begin{aligned} a^2 \approx & \alpha_{21}^2 [ (s_{12}\sigma_{11})^2 + (s_{22}\sigma_{21})^2 + (s_{32}\sigma_{31})^2 + (s_{11}\sigma_{12})^2 + (s_{21}\sigma_{22})^2 + (s_{31}\sigma_{32})^2 \\ & + 2(s_{12}\sigma_{11})(s_{22}\sigma_{21}) + 2(s_{12}\sigma_{11})(s_{32}\sigma_{31}) + 2(s_{12}\sigma_{11})(s_{11}\sigma_{12}) \\ & + 2(s_{12}\sigma_{11})(s_{21}\sigma_{22}) + 2(s_{12}\sigma_{11})(s_{31}\sigma_{32}) + 2(s_{22}\sigma_{21})(s_{32}\sigma_{31}) \\ & + 2(s_{22}\sigma_{21})(s_{11}\sigma_{12}) + 2(s_{22}\sigma_{21})(s_{21}\sigma_{22}) + 2(s_{22}\sigma_{21})(s_{31}\sigma_{32}) \\ & + 2(s_{32}\sigma_{31})(s_{11}\sigma_{12}) + 2(s_{32}\sigma_{31})(s_{21}\sigma_{22}) + 2(s_{32}\sigma_{31})(s_{31}\sigma_{32}) \\ & + 2(s_{11}\sigma_{12})(s_{21}\sigma_{22}) + 2(s_{11}\sigma_{12})(s_{31}\sigma_{32}) + 2(s_{21}\sigma_{22})(s_{31}\sigma_{32}) ] . \end{aligned} \quad (5.98)$$

For each  $s_{nm}\sigma_{n'm'}$  term, the variable of interest is  $\Delta\mathbf{y}$ . Thus, we can treat the  $s_{nm}$  portion as a

constant, meaning

$$\langle s_{nm}^2 \sigma_{n'm'}^2 \rangle = s_{nm}^2 \langle \sigma_{n'm'}^2 \rangle \quad (5.99)$$

Using the result from the Appendix and by substitution from Equation 5.98,  $\langle a^2 \rangle$  is

$$\begin{aligned} \langle a^2 \rangle &\approx \alpha_{21}^2 \langle (s_{12}\sigma_{11})^2 + (s_{22}\sigma_{21})^2 + (s_{32}\sigma_{31})^2 + (s_{11}\sigma_{12})^2 + (s_{21}\sigma_{22})^2 + (s_{31}\sigma_{32})^2 \\ &\quad + 2s_{12}s_{22}\sigma_{11}\sigma_{21} + 2s_{12}s_{32}\sigma_{11}\sigma_{31} + 2s_{12}s_{11}\sigma_{11}\sigma_{12} + 2s_{22}s_{32}\sigma_{21}\sigma_{31} + 2s_{21}s_{22}\sigma_{21}\sigma_{22} \\ &\quad + 2s_{31}s_{32}\sigma_{31}\sigma_{32} + 2s_{11}s_{12}\sigma_{21}\sigma_{22} + 2s_{11}s_{12}\sigma_{31}\sigma_{32} + 2s_{21}s_{22}\sigma_{31}\sigma_{32} \rangle \\ &\approx \alpha_{21}^2 s_{12}^2 \langle \sigma_{11}^2 \rangle + \alpha_{21}^2 s_{22}^2 \langle \sigma_{21}^2 \rangle + \alpha_{21}^2 s_{32}^2 \langle \sigma_{31}^2 \rangle + \alpha_{21}^2 s_{11}^2 \langle \sigma_{12}^2 \rangle \\ &\quad + \alpha_{21}^2 s_{21}^2 \langle \sigma_{22}^2 \rangle + \alpha_{21}^2 s_{31}^2 \langle \sigma_{32}^2 \rangle + 2\alpha_{21}^2 s_{12}s_{22} \langle \sigma_{11}\sigma_{21} \rangle + 2\alpha_{21}^2 s_{12}s_{32} \langle \sigma_{11}\sigma_{31} \rangle \\ &\quad + 2\alpha_{21}^2 s_{12}s_{11} \langle \sigma_{11}\sigma_{12} \rangle + 2\alpha_{21}^2 s_{22}s_{32} \langle \sigma_{21}\sigma_{31} \rangle + 2\alpha_{21}^2 s_{21}s_{22} \langle \sigma_{21}\sigma_{22} \rangle \\ &\quad + 2\alpha_{21}^2 s_{31}s_{32} \langle \sigma_{31}\sigma_{32} \rangle + 2\alpha_{21}^2 s_{11}s_{12} \langle \sigma_{21}\sigma_{22} \rangle + 2\alpha_{21}^2 s_{11}s_{12} \langle \sigma_{31}\sigma_{32} \rangle \\ &\quad + 2\alpha_{21}^2 s_{21}s_{22} \langle \sigma_{31}\sigma_{32} \rangle \end{aligned} \quad (5.100)$$

Similar expressions can be defined for  $\langle b^2 \rangle$  and  $\langle c^2 \rangle$ . If we expand the term  $\langle \sigma_{11}^2 \rangle$  from Equation 5.100, we find

$$\begin{aligned} \langle \sigma_{11}^2 \rangle &\approx \left\langle \left[ \sum_{k=1}^K w_k (\Delta \widetilde{x}_{kn} \widetilde{y}_{km} + \widetilde{x}_{kn} \Delta \widetilde{y}_{km}) \right]^2 \right\rangle \\ &\approx \left\langle [w_1 (\Delta \widetilde{x}_{11} \widetilde{y}_{11} + \widetilde{x}_{11} \Delta \widetilde{y}_{11}) + \dots + w_k (\Delta \widetilde{x}_{k1} \widetilde{y}_{k1} + \widetilde{x}_{k1} \Delta \widetilde{y}_{k1})]^2 \right\rangle. \end{aligned} \quad (5.101)$$

The  $\Delta \mathbf{x}_k$  represent the errors in determining the positions of the markers in the local frame. We consider these to be zero-mean, Gaussian random variables as well. Expanding Equation 5.101, we find that the mixed polynomial terms become zero, resulting in

$$\begin{aligned} \langle \sigma_{11}^2 \rangle &\approx \langle w_1^2 \widetilde{x}_{11}^2 \Delta \widetilde{y}_{11}^2 + \dots + w_k^2 \widetilde{x}_{k1}^2 \Delta \widetilde{y}_{k1}^2 \rangle \\ &\quad + \langle w_1^2 \Delta \widetilde{x}_{11}^2 \widetilde{y}_{11}^2 + \dots + w_k^2 \Delta \widetilde{x}_{k1}^2 \widetilde{y}_{k1}^2 \rangle \end{aligned}$$

$$\begin{aligned}
&\approx \langle w_1^2 \widetilde{x}_{11}^2 \Delta \widetilde{y}_{11}^2 \rangle + \dots + \langle w_k^2 \widetilde{x}_{k1}^2 \Delta \widetilde{y}_{k1}^2 \rangle \\
&\quad + \langle w_1^2 \Delta \widetilde{x}_{11}^2 \widetilde{y}_{11}^2 \rangle + \dots + \langle w_k^2 \Delta \widetilde{x}_{k1}^2 \widetilde{y}_{k1}^2 \rangle \\
&\approx w_1^2 \widetilde{x}_{11}^2 \langle \Delta \widetilde{y}_{11}^2 \rangle + \dots + w_k^2 \widetilde{x}_{k1}^2 \langle \Delta \widetilde{y}_{k1}^2 \rangle \\
&\quad + w_1^2 \widetilde{y}_{11}^2 \langle \Delta \widetilde{x}_{11}^2 \rangle + \dots + w_k^2 \widetilde{y}_{k1}^2 \langle \Delta \widetilde{x}_{k1}^2 \rangle \quad . \quad (5.102)
\end{aligned}$$

The general expression for  $\langle \sigma_{nm}^2 \rangle$  can therefore be expressed as

$$\begin{aligned}
\langle \sigma_{nm}^2 \rangle &\approx w_1^2 \widetilde{x}_{1n}^2 \langle \Delta \widetilde{y}_{1m}^2 \rangle + \dots + w_k^2 \widetilde{x}_{kn}^2 \langle \Delta \widetilde{y}_{km}^2 \rangle \\
&\quad + w_1^2 \widetilde{y}_{1m}^2 \langle \Delta \widetilde{x}_{1n}^2 \rangle + \dots + w_k^2 \widetilde{y}_{km}^2 \langle \Delta \widetilde{x}_{kn}^2 \rangle \quad . \quad (5.103)
\end{aligned}$$

This general expression for  $\sigma_{nm}$  indicates the influence of  $\Delta \mathbf{y}_k$  on the eigenvalues of the  $\mathbf{A}$  and  $\mathbf{B}$  matrices. The expanded  $a^2$  term (from Equation 5.98) is influenced by the first two components of  $\Delta \mathbf{y}_k$ , that is  $\Delta \mathbf{y}_{k1,2}$ . The  $b^2$  and  $c^2$  terms are influenced by  $\Delta \mathbf{y}_{k1,3}$  and  $\Delta \mathbf{y}_{k2,3}$ , respectively. As a result,  $\langle a^2 + b^2 + c^2 \rangle$  will change proportionally as  $\langle \Delta \mathbf{y}_k^2 \rangle$ , which indicates a change in  $\| \Delta \mathbf{y}_k \|$ . Therefore, we can conclude that if  $\| \Delta \mathbf{y}_k \|$  increases or decreases,  $|\lambda_{1,2}|$  increase or decrease, respectively. The change in eigenvalues indicates a proportional change in  $\| \mathbf{A} \|_2$  and  $\| \mathbf{B} \|_2$ , where  $\| \mathbf{A} \|_2$  is the norm-2, or matrix norm, of  $\mathbf{A}$  given by

$$\| \mathbf{A} \|_2 = (\lambda_{max}(\mathbf{A}^T \mathbf{A}))^{1/2} \quad , \quad (5.104)$$

with  $\lambda_{max}$  being the largest eigenvalue of  $\mathbf{A}$ . The increase in the matrix norms signal an increase in pose error. The expression for  $\sigma_{nm}$  also indicates the influence of  $\Delta \mathbf{x}_k$  on the eigenvalues of  $\mathbf{A}$  and  $\mathbf{B}$ . Similar to  $\Delta \mathbf{y}_k$ , a change in  $\langle \Delta \mathbf{x}_k^2 \rangle$  will proportionately affect the eigenvalues. However, the  $\Delta \mathbf{x}_k$  are generally smaller than the  $\Delta \mathbf{y}_k$ .

Because the  $\Delta\mathbf{y}_k$  and  $\Delta\mathbf{x}_k$  are random variables, we find that as the standard deviation of these quantities increases or decreases,  $\langle \Delta\mathbf{y}_k^2 \rangle$  increases or decreases indicating a proportional increase or decrease in  $\|\Delta\mathbf{y}_k\|$  and  $\|\Delta\mathbf{x}_k\|$ . The standard deviation of  $\Delta\mathbf{y}_k$  represents the accuracy of the tracking system and the standard deviation of  $\Delta\mathbf{x}_k$  indicates the error in determining the local marker locations. Thus, if the tracker and/or the local frame determination is more accurate, the standard deviation on  $\langle \Delta\mathbf{y}_k^2 \rangle$  is smaller, resulting in smaller eigenvalues for **A** and **B** and smaller pose errors. If the tracker and/or the local frame determination is less accurate, the standard deviation on  $\langle \Delta\mathbf{y}_k^2 \rangle$  is larger, yielding larger eigenvalues for **A** and **B** and, thereby, larger pose errors.

Another effect that we can observe from this formulation is that of the tracker precision, or repeatability. Theoretically, when we measure the ensemble average of a squared random variable with zero mean, we obtain the variance of its distribution. That is, if we assume that the tracker accuracy is 0.1 mm RMS, then by modeling  $\Delta\widetilde{\mathbf{y}}_k$  as a zero-mean, Gaussian random variable we expect  $\langle \Delta\widetilde{\mathbf{y}}_k^2 \rangle$  will be equal to  $(0.1\text{mm})^2 = 0.01\text{mm}^2$  theoretically. In practice, however, it is impossible to attain this value. Given a tracker with good precision, we can closely approximate this value with a large enough number of measurements. In contrast, a tracker with poor precision will attain a poor approximation, resulting in an increased variance and a larger standard deviation. A consequence of the larger standard deviation is that  $\Delta\widetilde{\mathbf{y}}_k$  must be increased to provide an accurate model of the tracking performance, which culminates in larger eigenvalues for **A** and **B** and, therefore, increased pose errors.

A formulation which encapsulates the eigenvalue magnitude is the matrix norm, as defined in Equation 5.104. In the case of an anti-symmetric matrix, both non-zero eigenvalues have the same magnitude. Therefore, the trends in pose estimation for a given tracking probe will be indicated by changes in the value of  $\|\mathbf{A}\|_2$ . If we simulate a logarithmic decrease in tracker accuracy ( $\Delta\mathbf{y}$  increases logarithmically), it can be shown that  $\|\mathbf{A}\|_2$  increases proportionally. The results of the simulation are shown in Figure 5.1.

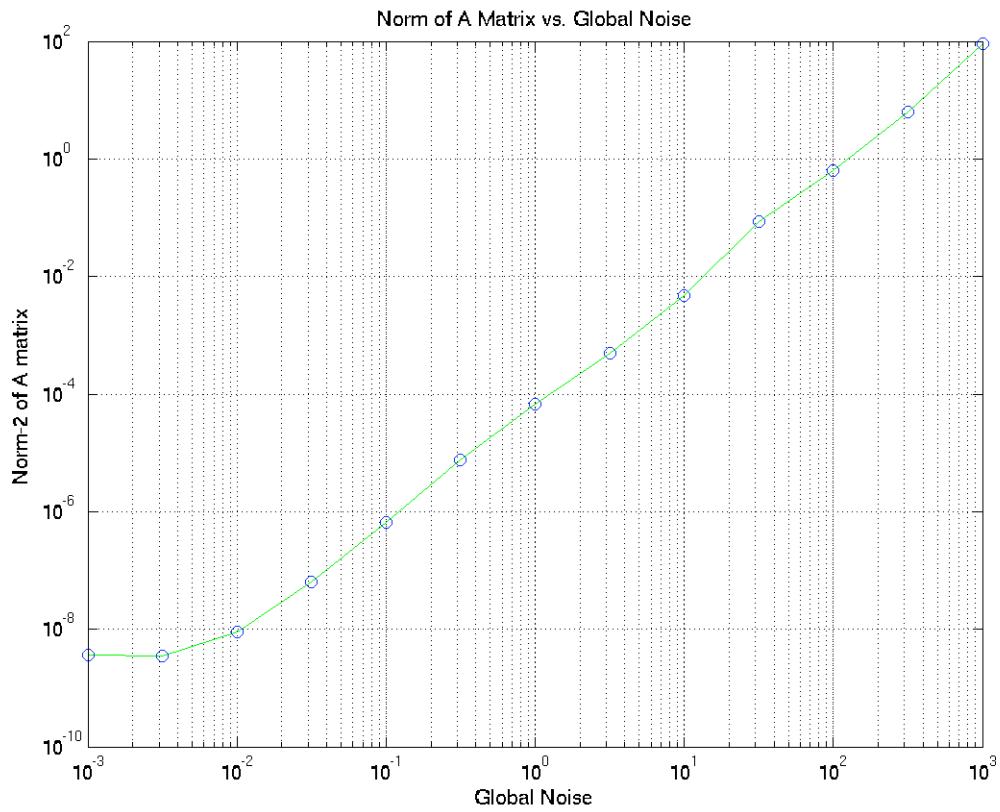


Figure 5.1: Semilog plot of the Norm of the A Matrix versus the Amount of Tracker Noise

### 5.2.2 The Effect of Probe Size and Shape on Pose Estimation

The size and shape of a tracking probe refer to its spatial extent and how the markers on the probe are arranged. Both are factors that have a significant effect upon pose estimation. To study these effects, we must determine the relationship between  $\widetilde{\mathbf{x}}_k$  and  $\widetilde{\mathbf{y}}_k$ , quantities which are defined by the probe size and shape, and the eigenvalues of  $\mathbf{A}$  and  $\mathbf{B}$ . Again, the change in the magnitude of the eigenvalues, whether the magnitude increases or decreases, is of interest.

We examine the effects of probe size and shape on the eigenvalues of  $\mathbf{A}$ , understanding that a similar formulation exists for  $\mathbf{B}$ . The influence of  $\widetilde{\mathbf{x}}$  and  $\widetilde{\mathbf{y}}$  can be seen in  $\alpha_{mn}$ ,  $s_{nm}$ , and  $\sigma_{nm}$ , which are present in the expressions of the eigenvalue components of  $\mathbf{A}$  (Equation 5.94). The  $\Delta \mathbf{y}_k$  and  $\Delta \mathbf{x}_k$  are also present, however, we shall hold their standard deviations constant, as we are narrowing our attention to the effects of probe size currently.

As expressed in Equation 5.94, the magnitude of the non-zero eigenvalues of  $\mathbf{A}$  increase or decrease as the squared magnitude increases or decreases. From Equation 5.100, we can see that all the variables previously mentioned affect the result. However, the most profound effect upon the value of  $\langle a^2 \rangle$  (or  $\langle b^2 \rangle$  or  $\langle c^2 \rangle$ ) is given by the  $\alpha_{mn}$  term, which is defined as

$$\alpha_{mn} = \frac{1}{\mu_m - \mu_n}, m \neq n, \quad (5.105)$$

where  $n$  is the row index of  $\mathbf{A}$ ,  $m$  is the column index of  $\mathbf{A}$ , and  $\mu_n$  and  $\mu_m$  correspond to the  $n^{th}$  and  $m^{th}$  eigenvalues of  $\mathbf{H}\mathbf{H}^T$ . The matrix  $\mathbf{H}$  is defined as

$$\mathbf{H} = \sum_{k=1}^K w_k \widetilde{\mathbf{x}}_k \widetilde{\mathbf{y}}_k^T, \quad (5.106)$$

where  $\widetilde{\mathbf{x}}_k$  and  $\widetilde{\mathbf{y}}_k$  represent the local and global marker locations relative to the marker centroid, respectively. From the dependence upon  $\widetilde{\mathbf{x}}_k$  and  $\widetilde{\mathbf{y}}_k$ , we see that the eigenvalues of  $\mathbf{H}\mathbf{H}^T$  will

change as the size of and the shape of the tracking probe changes. Furthermore, as the eigenvalues of  $\mathbf{H}\mathbf{H}^T$  change, the eigenvalues of  $\mathbf{A}$  and  $\mathbf{B}$  will change, as indicated by Equations 5.91 – 5.94 .

### 5.2.2.1 The Effects of Probe Size

Equation 5.100 provides the reasoning as to why  $\alpha_{mn}$  has the most influence upon  $\langle a^2 \rangle$  when changing the size and shape of a tracking probe. Changing the size of the tracking probe is equivalent to scaling the  $\mathbf{x}_k$  and  $\mathbf{y}_k$  by  $\beta$ , where  $\beta$  is a positive constant. Furthermore, a change in  $\mathbf{x}_k$  and  $\mathbf{y}_k$  will be proportionately expressed in  $\widetilde{\mathbf{x}}_k$  and  $\widetilde{\mathbf{y}}_k$ . Taking the case when  $\widetilde{\mathbf{x}}_k$  and  $\widetilde{\mathbf{y}}_k$  are scaled by  $\beta$ , we see that  $\sigma_{nm}$  becomes

$$\sigma_{nm} \approx \sum_{k=1}^K w_k (\Delta \widetilde{x}_{kn} (\beta \widetilde{y}_{km}) + (\beta \widetilde{x}_{kn}) \Delta \widetilde{y}_{km}) \quad (5.107)$$

$$\approx \beta \sum_{k=1}^K w_k (\Delta \widetilde{x}_{kn} \widetilde{y}_{km} + \widetilde{x}_{kn} \Delta \widetilde{y}_{km}) , \quad (5.108)$$

and that  $s_{nm}$  becomes

$$s_{nm} = \sum_{k=1}^K w_k (\beta \widetilde{x}_{kn}) (\beta \widetilde{y}_{km}) \quad (5.109)$$

$$= \beta^2 \sum_{k=1}^K w_k \widetilde{x}_{kn} \widetilde{y}_{km} . \quad (5.110)$$

From this relationship, we can see that the effect of changing the probe size by scaling  $\widetilde{\mathbf{x}}_k$  and  $\widetilde{\mathbf{y}}_k$  by  $\beta$  is a scaling of  $\mathbf{H}$  by  $\beta^2$  effectively. That is

$$\mathbf{H} = \begin{bmatrix} s_{11} & s_{12} & s_{13} \\ s_{21} & s_{22} & s_{23} \\ s_{31} & s_{32} & s_{33} \end{bmatrix} \rightarrow \begin{bmatrix} \beta^2 s_{11} & \beta^2 s_{12} & \beta^2 s_{13} \\ \beta^2 s_{21} & \beta^2 s_{22} & \beta^2 s_{23} \\ \beta^2 s_{31} & \beta^2 s_{32} & \beta^2 s_{33} \end{bmatrix} \rightarrow \beta^2 \mathbf{H} . \quad (5.111)$$



Also, scaling  $\widetilde{\mathbf{x}}_k$  and  $\widetilde{\mathbf{y}}_k$  by  $\beta$  scales  $\Delta\mathbf{H}$  by  $\beta$ , giving

$$\Delta\mathbf{H} \approx \beta \sum_{k=1}^K w_k (\Delta\widetilde{\mathbf{x}}_k \widetilde{\mathbf{y}}_k^T + \widetilde{\mathbf{x}}_k \Delta\widetilde{\mathbf{y}}_k^T) \rightarrow \beta \Delta\mathbf{H} . \quad (5.112)$$

Scaling  $\mathbf{H}$  and  $\Delta\mathbf{H}$  results in a scaling of  $\Delta\mathbf{Q}$  by  $\beta^3$ , that is

$$\Delta\mathbf{Q} \approx \Delta\mathbf{H}^T \mathbf{H} + \mathbf{H}^T \Delta\mathbf{H} \rightarrow \beta \Delta\mathbf{H}^T \beta^2 \mathbf{H} + \beta^2 \mathbf{H}^T \beta \Delta\mathbf{H} \quad (5.113)$$

$$\rightarrow \beta^3 (\Delta\mathbf{H}^T \mathbf{H} + \mathbf{H}^T \Delta\mathbf{H}) \rightarrow \beta^3 \Delta\mathbf{Q} . \quad (5.114)$$

The scaling also results in

$$\mathbf{H}^T \mathbf{H} \rightarrow \beta^4 \mathbf{H}^T \mathbf{H} , \quad (5.115)$$

implying that

$$\mu_m \rightarrow \beta^4 \mu_m , \quad (5.116)$$

while  $\mathbf{v}_m$  remains the same due to the fact that it is a normalized eigenvector. The effect of the scaling is then related to the components of the  $\mathbf{A}$  matrix as

$$a = \alpha_{21} \mathbf{v}_1^T \Delta\mathbf{Q} \mathbf{v}_2 \rightarrow \frac{\mathbf{v}_1^T \beta^3 \Delta\mathbf{Q} \mathbf{v}_2}{\beta^4 (\mu_2 - \mu_1)} \rightarrow \frac{a}{\beta} \quad (5.117)$$

$$b \rightarrow \frac{b}{\beta} \quad (5.118)$$

$$c \rightarrow \frac{c}{\beta} . \quad (5.119)$$

Therefore, scaling  $\widetilde{\mathbf{x}}_k$  and  $\widetilde{\mathbf{y}}_k$  by  $\beta$  results in an effective scaling of approximately  $1/\beta$  for the eigenvalues of  $\mathbf{A}$ . A similar formulation can be demonstrated for  $\mathbf{B}$ . Equations 5.117 – 5.119 show that increasing or decreasing the size of a tracking probe results in a respective improvement or reduction in pose error for the tracking probe.

Again, using  $\|\mathbf{A}\|_2$ , we can demonstrate the effects of increased probe size on pose estimation. If we simulate a logarithmic increase in probe size ( $\mathbf{x}_k$  are uniformly scaled with logarithmically

increasing scale factors), it can be shown that  $\| \mathbf{A} \|_2$  decreases proportionally. The results of the simulation are shown in Figure 5.2.

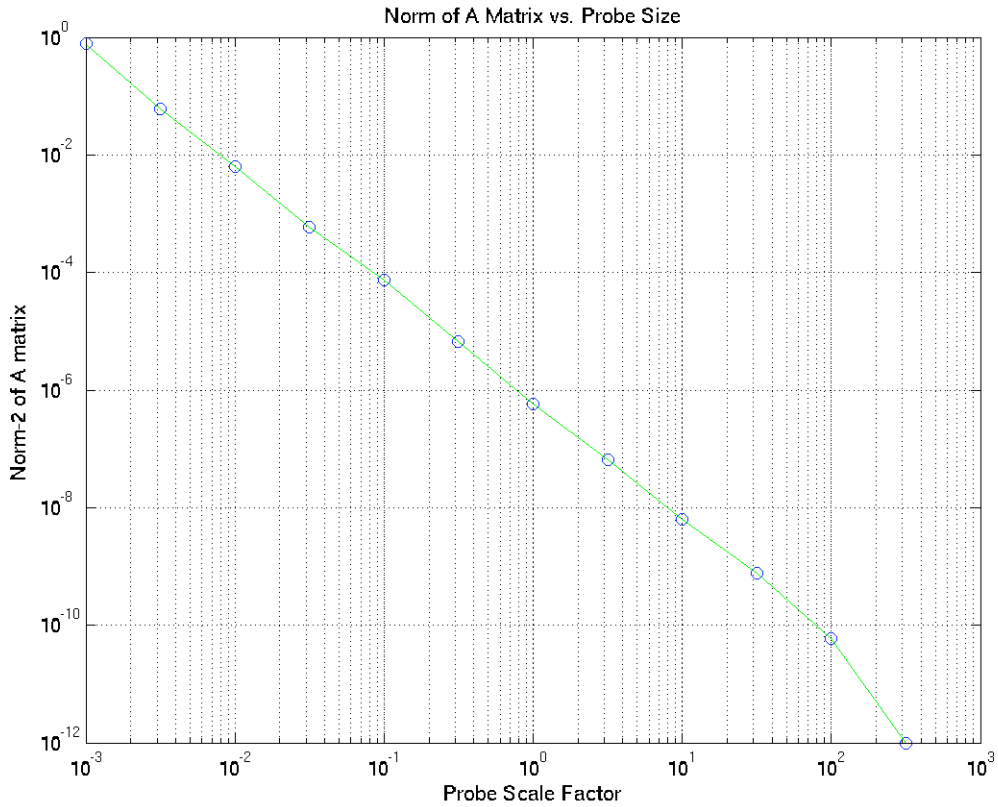


Figure 5.2: Semilog plot of the Norm of the A Matrix versus the Size of the Probe

### 5.2.2.2 The Effects of Probe Shape

A change in the size of a tracking probe indicates uniform scaling applied to the  $\mathbf{x}_k$  and  $\mathbf{y}_k$ . Changing the probe shape, however, implies that a different  $\mathbf{x}_k$  and  $\mathbf{y}_k$  are used. Assuming that one

marker is moved at a time, we can represent the different  $\mathbf{x}_k$  and  $\mathbf{y}_k$  as the resultant of a sum of vectors, meaning

$$\mathbf{x}_k' = \mathbf{x}_k + \delta\mathbf{x}_k \text{ and} \quad (5.120)$$

$$\mathbf{y}_k' = \mathbf{y}_k + \delta\mathbf{y}_k , \quad (5.121)$$

where  $\delta\mathbf{x}_k$  and  $\delta\mathbf{y}_k$  are the change applied to  $\mathbf{x}_k$  and  $\mathbf{y}_k$ , respectively. The change in the given marker will also influence  $\widetilde{\mathbf{x}}_k$  and  $\widetilde{\mathbf{y}}_k$  for all the markers on the probe, as these variables are dependent upon the centroid of the markers. Therefore, assuming that the  $\Delta\widetilde{\mathbf{x}}_k$  and  $\Delta\widetilde{\mathbf{y}}_k$  remain the same, we obtain

$$\widetilde{\mathbf{x}}_k' = (\mathbf{x}_k - \mathbf{x}') + \Delta\widetilde{\mathbf{x}}_k \text{ and} \quad (5.122)$$

$$\widetilde{\mathbf{y}}_k' = (\mathbf{y}_k - \mathbf{y}') + \Delta\widetilde{\mathbf{y}}_k , \quad (5.123)$$

where  $\mathbf{x}'$  and  $\mathbf{y}'$  are the centroids of the markers in the local and global frames, taking into account the marker location which has changed.

The quantity  $\alpha$  will still have the most influence upon the eigenvalues of  $\mathbf{A}$ . We examine the relationship for  $\langle a^2 \rangle$ , taking the case when  $\widetilde{\mathbf{x}}_k$  and  $\widetilde{\mathbf{y}}_k$  are changed by  $\delta\mathbf{x}_k$  and  $\delta\mathbf{y}_k$ . As a result,  $\sigma_{nm}$  now becomes

$$\sigma_{nm} \approx \sum_{k=1}^K w_k (\Delta\widetilde{x}_{kn}\widetilde{y}_{km}' + \widetilde{x}_{kn}'\Delta\widetilde{y}_{km}) , \quad (5.124)$$

and  $s_{nm}$  becomes

$$s_{nm} = \sum_{k=1}^K w_k \widetilde{x}_{kn}' \widetilde{y}_{km}' . \quad (5.125)$$

Because we assume that  $\mathbf{x}_k \gg \Delta\mathbf{x}_k$  and  $\mathbf{y}_k \gg \Delta\mathbf{y}_k$ , it can be shown that  $\widetilde{x}_{kn}'\widetilde{y}_{km}' > \Delta\widetilde{x}_{kn}\widetilde{y}_{km}' + \widetilde{x}_{kn}'\Delta\widetilde{y}_{km}$ . Thus, the effect of changing the probe shape has a greater effect on the  $s_{nm}$  terms within the expression for  $\langle a^2 \rangle$ .

This formulation provides a relationship between the individual marker locations and the eigenvalues of  $\mathbf{A}$ , which, in turn, defines the relationship between the elements of  $\mathbf{A}$  and the amount of pose error present from a given tracking probe topology. A proposed quantitative measure of this relationship is explored in Section 7.6.

### 5.2.3 The Effect of the Number of Markers and Marker Field of Emission on Pose Estimation

Another effect to examine is that of the number of markers on the tracking probe and its effect on pose estimation. If we start with the expression for pose error,

$$\epsilon(\mathbf{R}, \mathbf{t}) = \sum_{k=1}^K w_k \| \mathbf{y}_k - \mathbf{R}\mathbf{x}_k - \mathbf{t} \|^2, \quad (5.126)$$

we see that  $\epsilon$  does not obviously increase or decrease if  $K$  changes. Also, examining following expressions

$$s_{nm} = \sum_{k=1}^K w_k \widetilde{x}_{kn} \widetilde{y}_{km} \quad \text{and} \quad (5.127)$$

$$\sigma_{nm} \approx \sum_{k=1}^K w_k (\Delta \widetilde{x}_{kn} \widetilde{y}_{km} + \widetilde{x}_{kn} \Delta \widetilde{y}_{km}), \quad (5.128)$$

we observe that neither  $s_{nm}$  or  $\sigma_{nm}$  necessarily increase or decrease as  $K$  changes.

There is no immediately apparent relationship between the number of markers on the probe and the accuracy of pose estimation. Nevertheless, insight may be obtained from the realization that the overall error in pose is expressed as a least-squares optimization problem. In a least-squares sense, data added to the approximation will either bring it closer to an optimal solution or drive it further away. The dimensions of a rigid tracking probe are fixed, however, meaning the data (the marker locations) are physically restricted to a given space. Thus, increasing the number of markers adds to the data in such a way as to drive the approximation closer to an optimal solution, giving a reduction in  $\epsilon$ . However, the additional number of markers required to see a significant change

may be unrealistic. The pose error of a tracking probe whose number of markers are increased dramatically is modeled in Section 7.1. An added benefit to increasing the number of markers may be an increase the field of regard of the tracking probe.

Trackers that use active markers are also affected by the marker field of emission. The larger the marker field of emission, the fewer number of markers that are required to achieve a large field of regard. As the solid angle of marker signal emission increases, the probe field of regard will likely increase. As the field of emission is generally fixed for a tracker, however, field of regard is more dependent upon number of markers used on the tracking probe.

#### 5.2.4 Matrix of Tracking Factors

The results of the previous sections can be summarized as a table of tracking factors versus performance in various aspects of pose estimation.

Table 5.1: Matrix of Tracking Factors

		pose error	rotation error	translation error	field of regard
global error	increases	+	+	+	~
	decreases	-	-	-	~
local error	increases	+	+	+	~
	decreases	-	-	-	~
probe size	increases	-	-	~	~
	decreases	+	+	~	~
number of markers	increases	+	~	~	+
	decreases	-	~	~	-
field of emission	increases	~	~	~	+
	decreases	~	~	~	-

The implications of Table 5.1 are that we can quickly determine what the effects of changing a particular parameter will be when designing a tracking probe. A “+” indicates an increase, a “-” indicates decrease, and “~” indicates no significant change in the measure of performance.

Combined with quantitative measures of performance, we can more effectively design and implement marker-based tracking probes using the techniques at our disposal.

## CHAPTER 6: APPLICATION TO TRACKING PROBES

By combining design methods, mathematical formulations, and simulation, we have created a framework for implementing marker-based tracking probes. We now discuss the application of the methods discussed in Chapters 3, 4, and 5 .

To test the implications of the framework, we use four probe topologies. The first tracking probe is a six marker digitizing probe manufactured by Northern Digital, Inc. The next probe consists of eight markers arranged in two concentric squares. The third probe is a semi-spherical, conformally designed tracking probe. The last probe is a conformally designed probe whose markers are mapped directly upon the object to be tracked.

The chapter begins with detail of the simulations of the tracking probes. Next, we provide experimental verification of the results from the simulations and the conformal probe design process. We close by detailing the design process of a conformal tracking probe.

### 6.1 Simulation of Tracking Probes

The simulation of tracking probes models the pose estimation process. The first step in this process is to determine the location of the markers within the local frame of the tracking probe,  $\mathbf{x}_k$ . The local marker positions are given by the marker mapping procedure, the schematics of the proposed tracking probe, or from user-created data files that specify a tracking probe. However, these locations are do not have any errors associated with them when produced by the mapping algorithms. For example, within the marker mapping procedure, we use a 3D model of an object that is composed of triangles to place the markers. When a marker location has been determined, it is placed at the center of the corresponding triangle. In practice, the marker location can only be approximated, usually to the size of the corresponding triangle within the 3D model. In addition, once the markers are placed on the object, the local coordinate frame may be changed, that is, recomputed based upon convenience. Therefore, we must add errors to these local marker locations to properly

simulate the construction of a tracking probe. The magnitude of the errors can be determined based upon manufacturing data (if the probe is already physically constructed) or based upon the relative size of the triangles used in the 3D model. The errors in determination of the local frame,  $\Delta\mathbf{x}_k$ , are simulated by applying zero-mean Gaussian noise to the coordinates using the following procedure:

1. Determine the amount of noise to apply to local coordinates.
2. Seed the random number generator.
3. Using three uniform random numbers, determine a normalized noise vector for each marker.
4. Scale the normalized noise vector magnitude using Gaussian random number. The Gaussian distribution used has zero-mean and a standard deviation equal to the noise amount chosen in Step 1.
5. Add the noise vectors to local marker locations.

Once we have simulated the local coordinates, we simulate the locations of the markers in the tracker frame of reference,  $\mathbf{y}_k$ . However, we must know where the probe is to be simulated within the tracking volume because the accuracy of the tracker may vary according to the probe location within the volume. Thus, we apply a rotation,  $\mathbf{R}$  and translation,  $\mathbf{t}$  to the local marker locations (with noise) to obtain the marker locations with respect to the tracker, also called the global marker locations. The rotation is specified in a matrix format, due to the formulation of the noise propagation procedure specified in Chapter 5. Finally, based upon the accuracy of the tracker in that region, we add zero-mean Gaussian noise,  $\Delta\mathbf{y}_k$  to the global marker locations using the procedure previously defined for the local marker coordinates.

We now have noisy and noise-free representations of the probe markers in the global and local coordinate frames. The next step is to estimate the amount of pose error. The procedure for propagating the effects of marker noise to the pose estimation is described in Section 5.1. An algorithm has been created and implemented to mimic this procedure. The inputs to the algorithm



are the local and global marker coordinates with noise, the standard deviation of the Gaussian noise applied to the local and global marker coordinates, and the initial, error-free rotation matrix that defines the orientation of the tracking probe. The output of the algorithm is the incremental rotation matrix,  $\Delta\mathbf{R}$  that transforms the tracking probe from its initial, error-free orientation to the orientation obtained when noisy marker data are used. From the  $\Delta\mathbf{R}$ , we compute the average pose error using a sum of squared errors formulation

$$\Delta P^2 = \frac{1}{K} \sum_{i=1}^K \|(\mathbf{R}_{error}\mathbf{x}_k + \mathbf{t}_{error}) - (\mathbf{R}\mathbf{x}_k + \mathbf{t})\|^2 . \quad (6.1)$$

In this expression,  $\Delta P$  is the pose error in millimeters,  $\mathbf{R}_{error}$  is the concatenation of the initial, error-free orientation and the incremental rotation matrix ( $\Delta\mathbf{R}\mathbf{R}$ ), the  $\mathbf{x}_k$  are the noise-free local marker locations, and  $\mathbf{t}_{error}$  is the sum of the initial position of the tracking probe and the translation between the noise-free local marker centroid and the noisy global marker centroid.

We can also determine the rotational and translational components of the pose error. The amount of error in rotation,  $\Delta P_{rot}$ , can be determined from a concatenation of two quaternions, expressed as

$$q_{result} = qq_{error}^{-1} , \quad (6.2)$$

where  $q$  is the unit quaternion representing the error-free orientation, obtained from  $\mathbf{R}$ , and  $q_{error}$  is the noisy orientation, obtained from  $\mathbf{R}_{error}$ . The amount of error in rotation is equal to the angle extracted from the scalar portion of  $q_{result}$ . The error in position,  $\Delta P_{pos}$ , is expressed as

$$\Delta P_{pos} = \|\mathbf{t}_{error} - \mathbf{t}\| . \quad (6.3)$$

In the course of the simulation process, we determine the value of  $\Delta P$ ,  $\Delta P_{rot}$ , and  $\Delta P_{pos}$

The independent variables within the simulation are the tracker noise, the probe size, and, the number of markers on the probe. The dependent variables we can study are the overall pose error in millimeters, the orientation error in degrees, and the position error in millimeters. The

tracker noise and probe size are incremented on a logarithmic scale. The number of markers is incremented arithmetically when a generic tracking probe shape is used. Currently, the simulation supports generic spherical and planar tracking probes with up to 25 markers.

At each increment of the independent variable, the probe is simulated  $N$  times ( $N \geq 20$ ) and the pose error results are averaged. The average at each increment is then plotted on a linear or log scale. When plotted on a linear scale, the standard deviation of the  $N$  analyses is included at each data point as error bars. Finally, a curve is fit to the data.

### *6.1.1 Simulation Specifics*

The simulation of tracking probes and pose errors was implemented in MATLAB. The simulation was built with a graphical user interface, to allow for easier simulation control and easier adjustment of simulation parameters. The simulation GUI is shown in Figure 6.1.

Aside from the dependent and independent variables, we can adjust the local noise of the probe markers, the increment applied to the independent variables, the number of iterations used to compute the average errors, how the results are displayed, and the initial probe position within the tracking volume.

The POLYFIT function is used to determine the coefficients of a polynomial curve that fits the data in the least squares sense. The POLYVAL function is used to obtain error estimates on the predictions from the polynomial. The error bounds assume that the data are independent, normally distributed, and with constant variance. The graphs are then generated with the PLOT and ERRORBAR functions. The green line corresponds to the results of a 4th order polynomial least-squares curve fit. The red lines above and below represent the 95% confidence interval for the data points plotted on the graph.

A 4th order polynomial chosen as opposed to an exponential function because a better fit is obtained when including the data for very small or very large values of the independent variable.

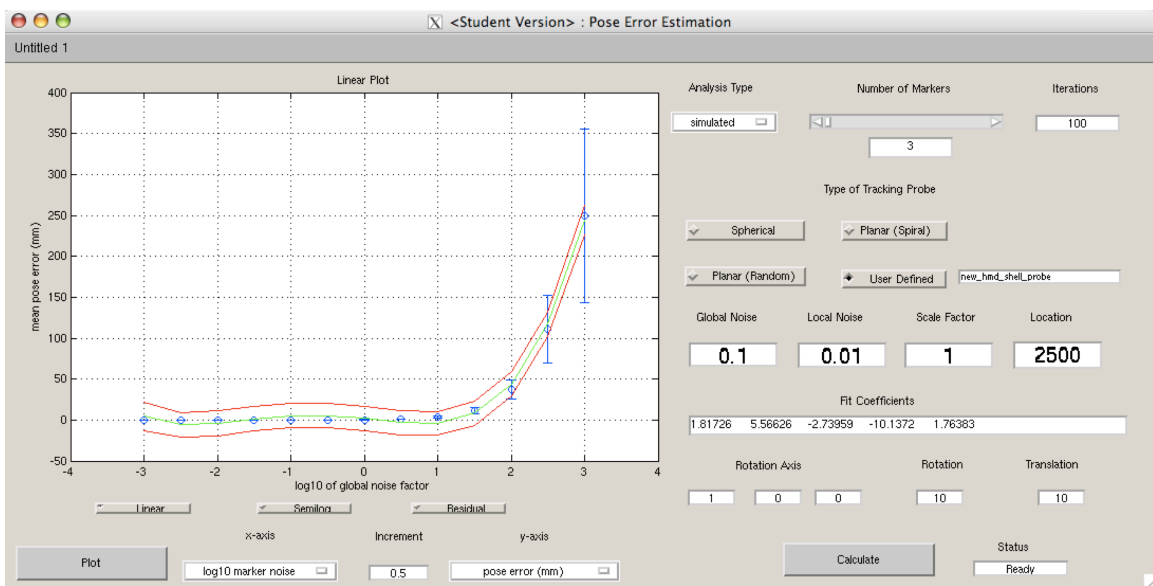


Figure 6.1: MATLAB Simulation GUI

Higher orders of magnitude were chosen for the purpose of displaying where the assumptions of the model are violated, that is, the order of magnitude of the noise is equal to that of the probe size. Still, the r-squared value for the 4th order curve fit ranges between 0.62 and 0.75 when a probe is simulated. Within two orders of magnitude of the midpoint, however, an exponential fit yields residuals on the order of 1 mm in pose error and r-squared values greater than 0.7.

Within the simulation we use the EXPM function to compute the matrix exponential. This function uses a Padé approximation rather than a Taylor Series approximation. We also use the RAND and RANDN functions to generate uniform and Gaussian random variables, respectively. The random number generators are seeded using the system time.

There are three types of generic tracking probes that are utilized in the simulation. The first type, a generic spherical probe, is composed of markers placed on a sphere of unit radius. The marker distributions are calculated beforehand using the modified SA technique described in Chapter 3. The next type is a square, 10 cm tracking probe with the markers arranged in a spiral fashion. That is, the probe is divided into a 5x5 grid with the markers arranged as numbered in Table 6.1.1.

The last type is a square, 10 cm probe with the markers placed randomly. These simulated probes

Table 6.1: Marker Placements for a Spiral, Planar Tracking Probe

15	16	17	18	19
14	4	5	6	20
13	3	1	7	21
12	2	9	8	22
11	10	25	24	23

were used as test cases for the probes to be physically implemented or as thought experiments to probe various aspects of the probe design framework.

## 6.2 Experimental Results from Tracking Probes

The model proposed in Chapter 5 relies upon the fact that the noise values propagated are Gaussian in nature. Therefore, our first task was to determine if this was the case experimentally. Using 1000 samples of position data from a single marker, a Gaussian curve was fit to a 50-bin histogram. The result of the fit is shown in Figure 6.2 and has an R-squared value of 0.9602, indicating a good approximation of a Gaussian distribution. The results show that the data are well fit by a Gaussian distribution, so our initial assumption of the noise distribution is valid. Moreover, the central limit theorem essentially states that data which are influenced by many small and unrelated random effects are approximately normally distributed.

Each tracking probe was simulated at a location of 2.5m from the tracker using local noise specifications which matched its physical tracking characteristics. The tracker accuracy simulated was 0.1 mm, with 100 iterations performed at each increment of the independent variable studied.

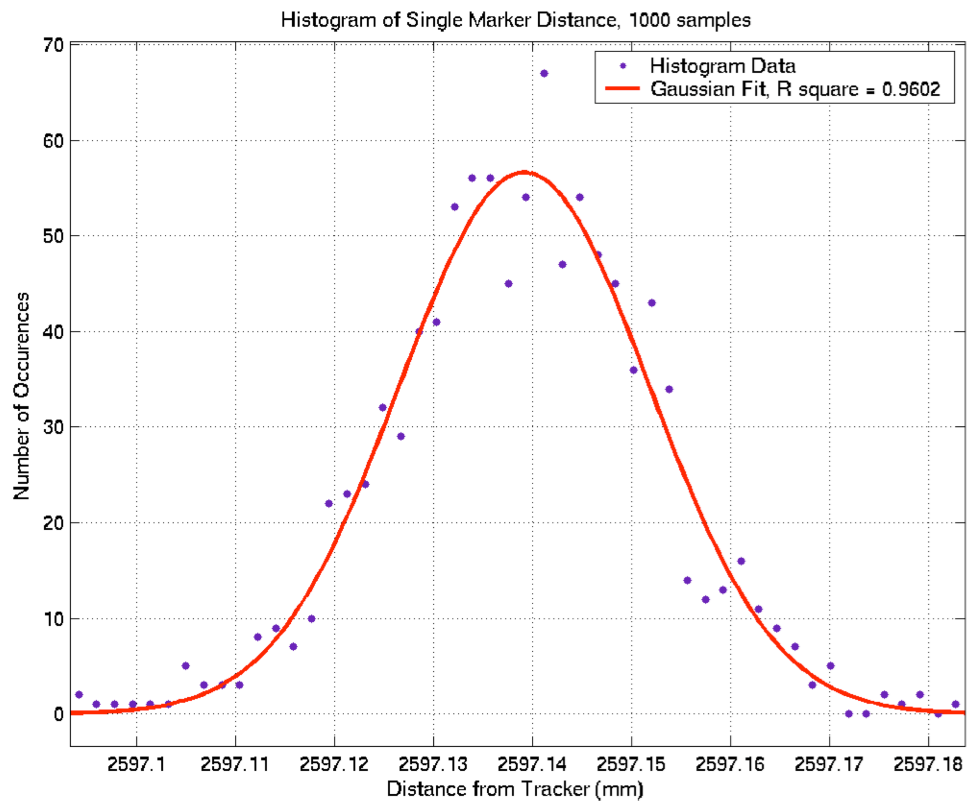


Figure 6.2: Data Points Fit with a Gaussian

For each tracking probe, we then performed an iterative accuracy measurement. The standard procedure was the following:

1. Secure probe to rotation stage
2. Record the unit quaternion describing the initial orientation of the probe, and the three element vector describing the initial position of the probe with respect to the tracker.
3. Rotate or translate the probe by a specified amount
4. Record the final orientation and position of the tracking probe
5. Determine the amount of rotation and translation reported by the tracker and compare

When measuring the accuracy in orientation, we use the error metric presented in Equation 6.2. Likewise, when determining accuracy in position, we use Equation 6.3

### 6.2.1 A Six-Marker Planar Probe

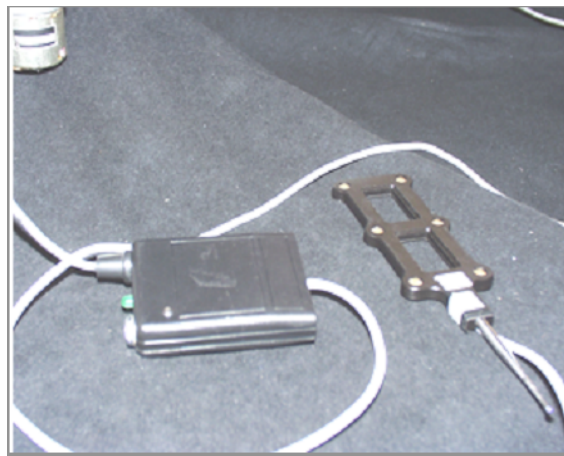


Figure 6.3: A Six-marker Planar Probe

The first tracking probe assessed was a planar probe with six markers, which corresponds to the digitizing probe shown in Figure 6.3. The local marker noise value for the probe was set at

0.127 mm to coincide with the manufacturing data for the accuracy of marker placement. The simulation of this probe predicted an error of  $0.06 \pm 0.04$  deg, shown in Figures 6.4 and 6.5. The black rectangle shown in Figure 6.4 indicates the area displayed in Figure 6.5. The experimental result was  $0.083 \pm 0.015$  deg.

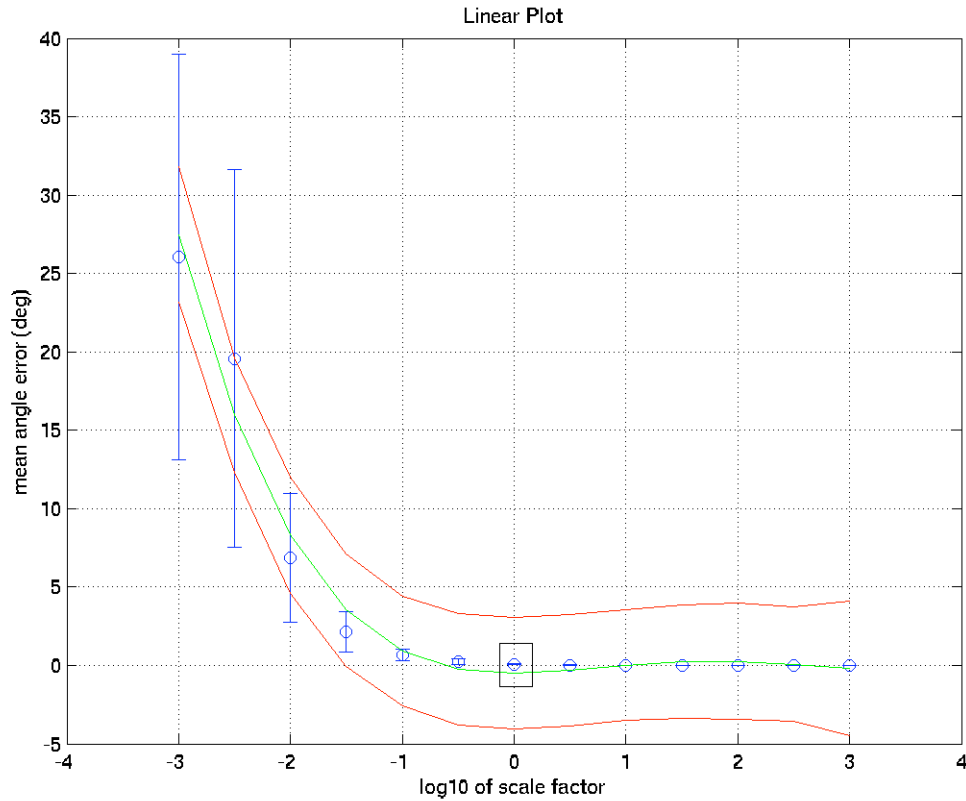


Figure 6.4: Simulation Results from a Six-Marker Planar Tracking Probe

### 6.2.2 Two Concentric Tracking Probes

To show the dependence of size on angular accuracy, we constructed two concentric tracking probes. The probes were rectangular and approximately at a ratio of 4:1 in size. The smaller probe consists of the four inside LEDs and has dimensions 3 cm x 2.5 cm. The large probe has dimensions 11.7 cm x 10.5 cm. In simulating the pair of probes, the local marker noise value for

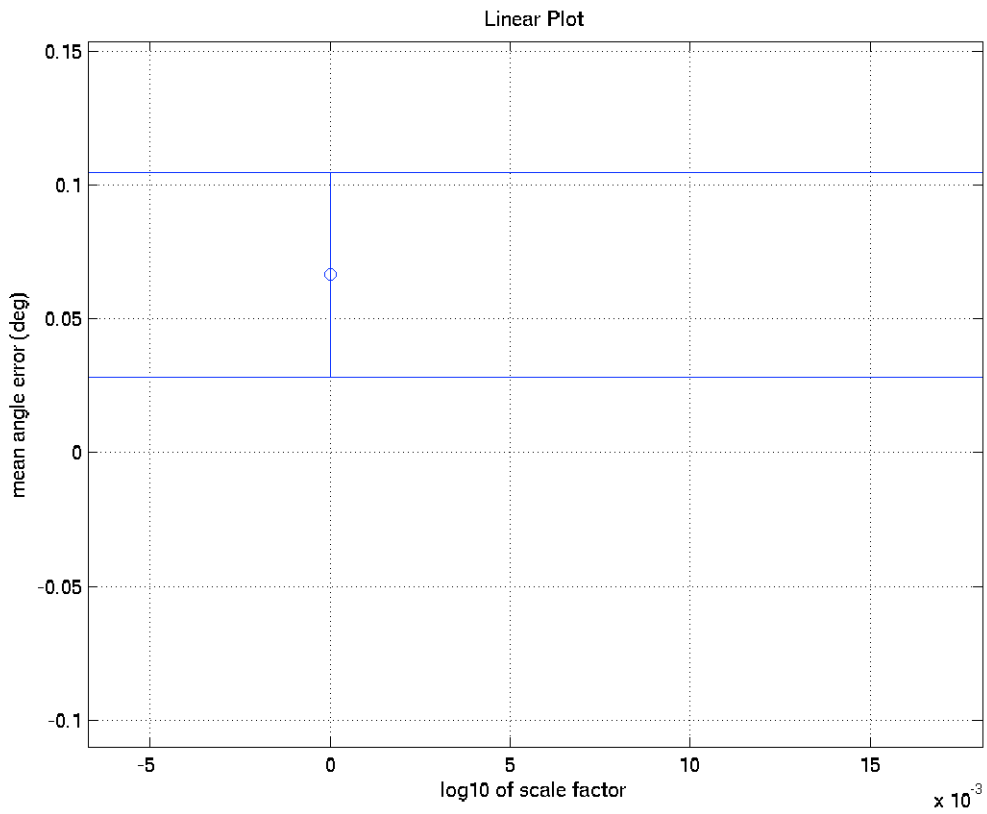


Figure 6.5: Simulation Results from a Six-Marker Planar Tracking Probe (at close range)



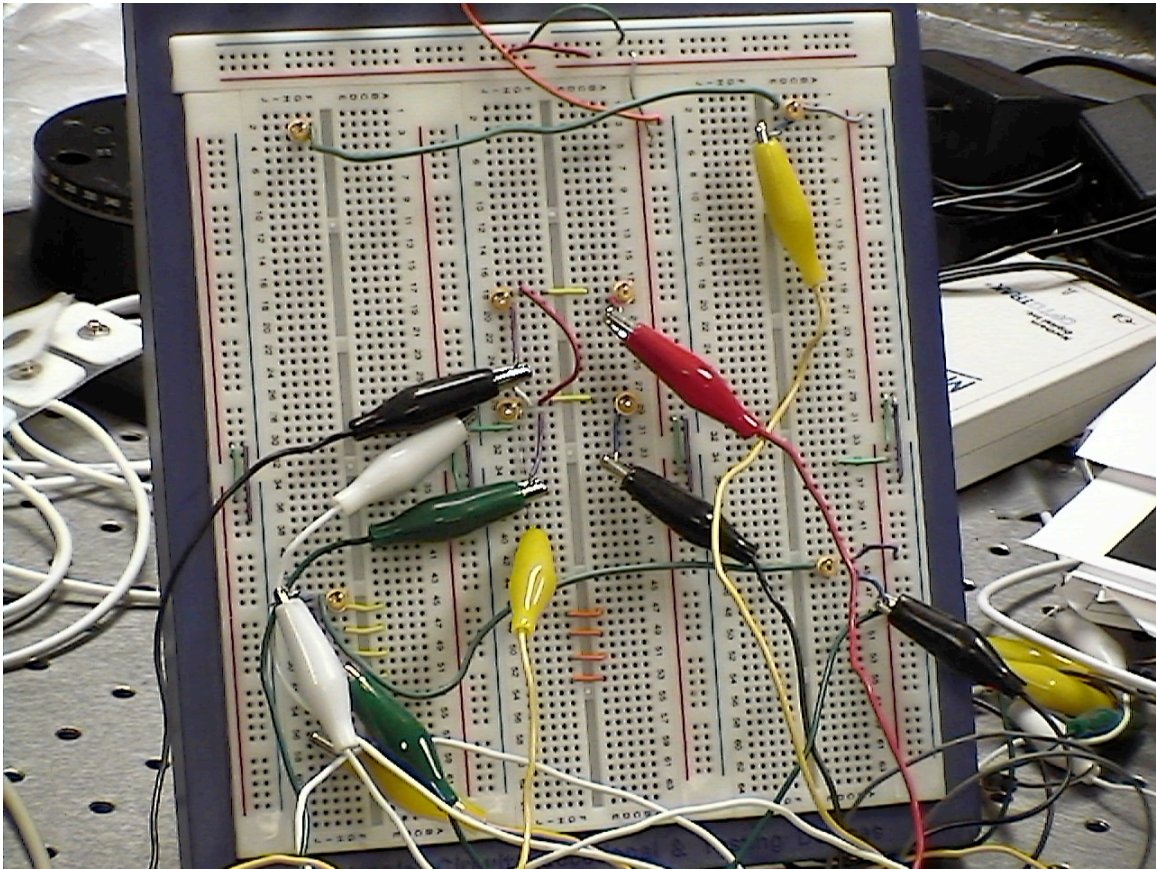


Figure 6.6: Two Concentric Tracking Probes

the probe was set at 3.5 mm to coincide with the distance between rows on the solderless breadboard. For the smaller tracking probe, the simulation predicted an orientation error of 0.206 deg, shown in Figures 6.7 and 6.8. The black rectangle shown in Figure 6.7 indicates the area displayed in Figure 6.8. The experimental result was  $0.282 \pm 0.03$  deg.

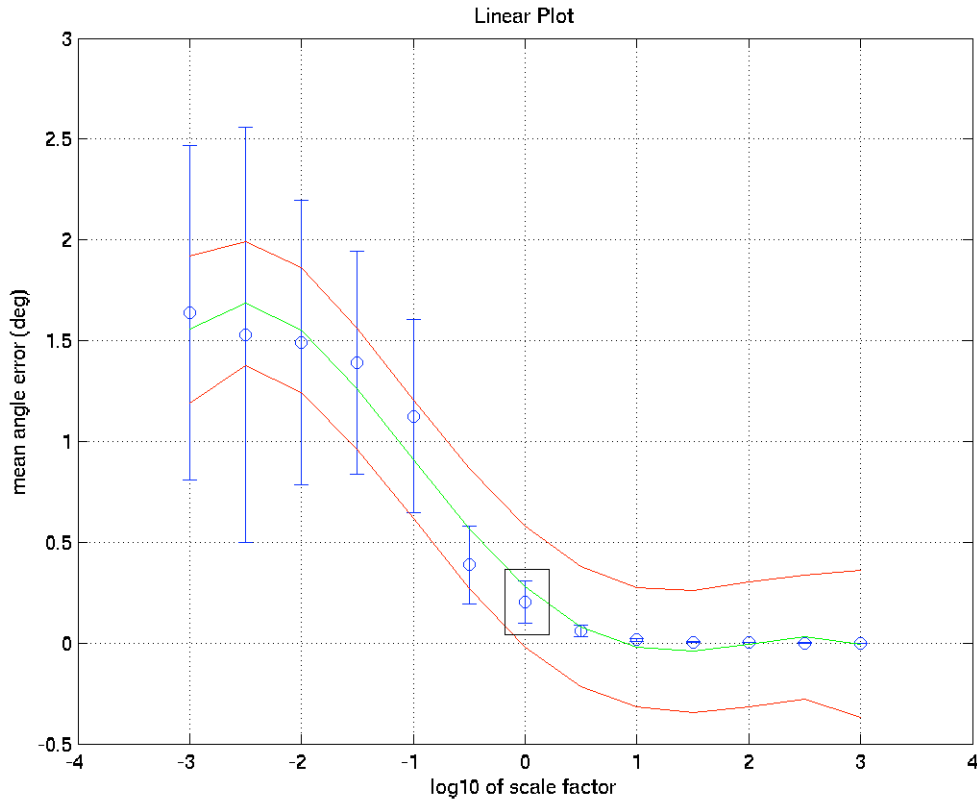


Figure 6.7: Simulation Results from Small Concentric Tracking Probe

Using the larger probe, the simulation predicted an orientation error of 0.039 deg, shown in Figures 6.9 and 6.10. The black rectangle shown in Figure 6.9 indicates the area displayed in Figure 6.9. The experimental result was  $0.008 \pm 0.026$  deg, Figure 6.6 shows the two tracking probes.

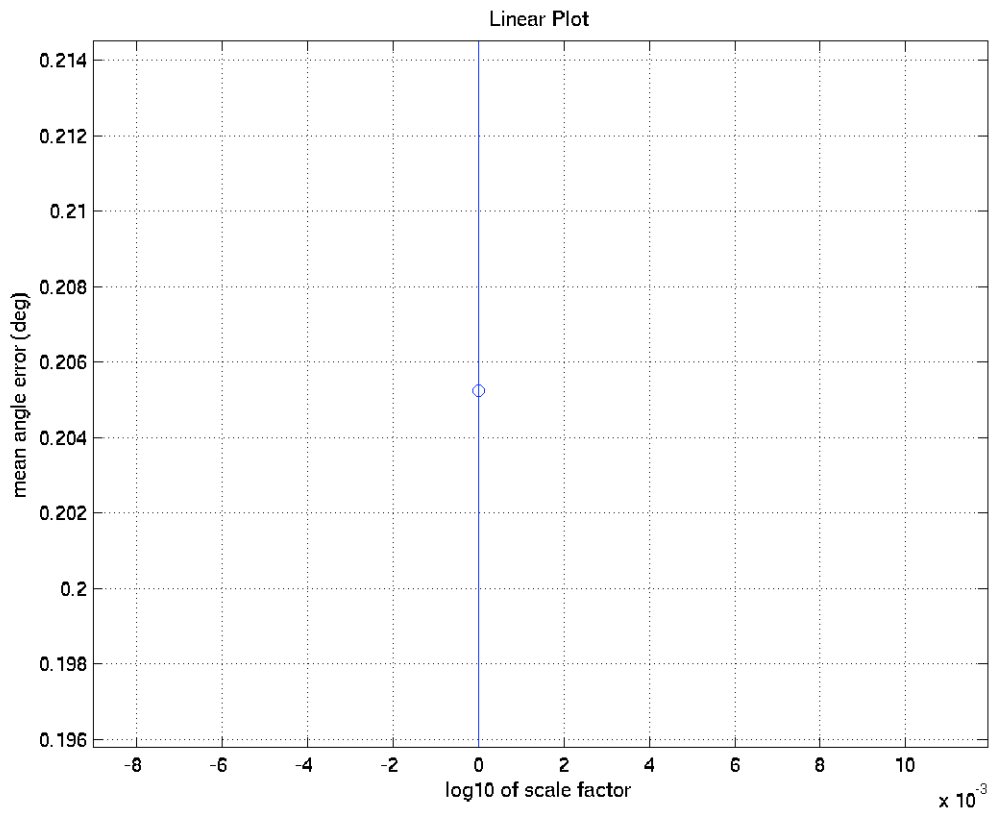


Figure 6.8: Simulation Results from Small Concentric Tracking Probe (at close range)

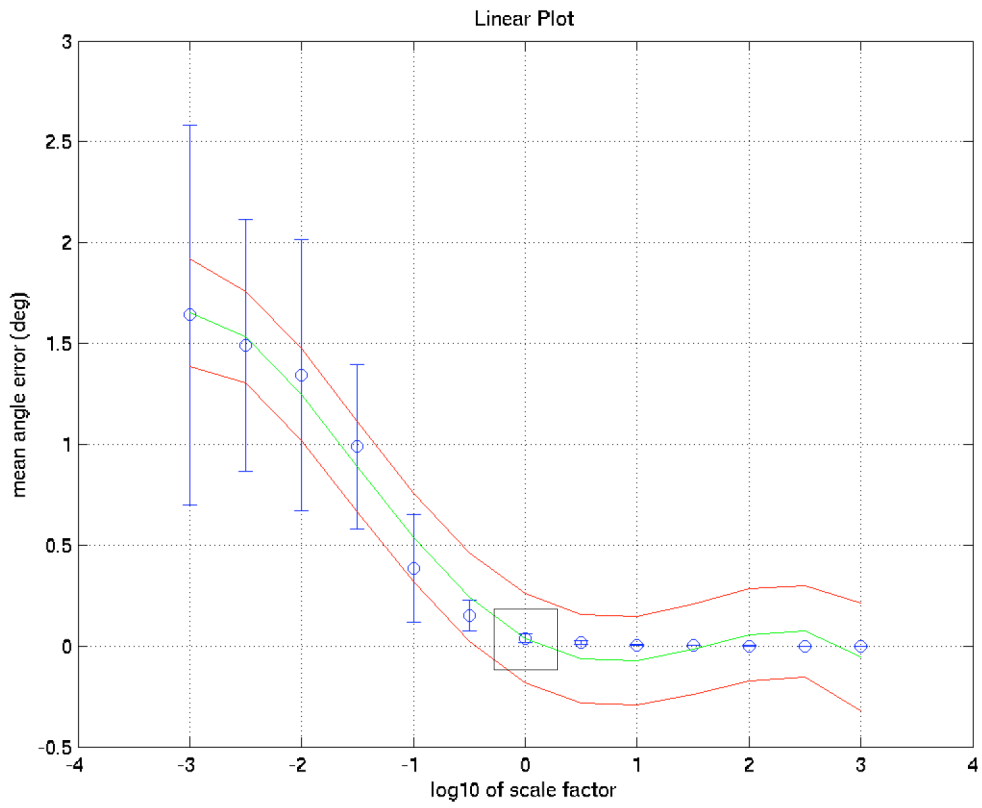


Figure 6.9: Simulation Results from Large Concentric Tracking Probe

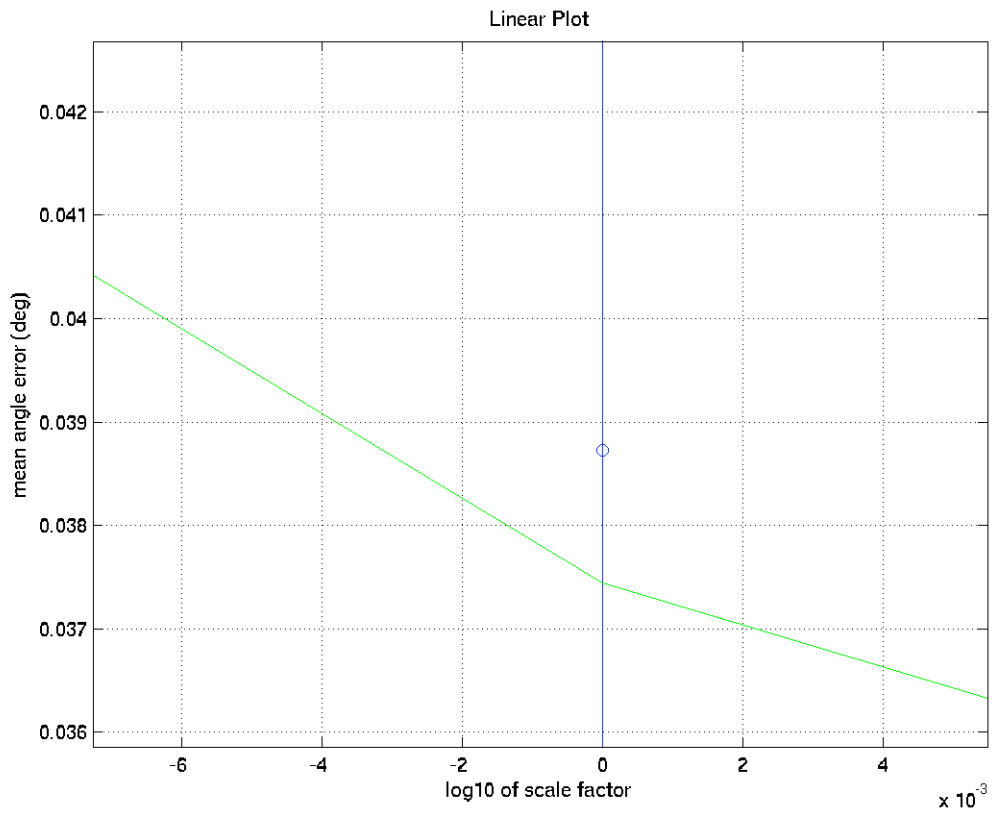


Figure 6.10: Simulation Results from Large Concentric Tracking Probe (at close range)

## 6.3 Case Study: Design, Implementation, and Assessment of Conformal Probes

### 6.3.1 A Semi-spherical Head Tracking Probe



Figure 6.11: A Semi-spherical Head Tracking Probe

The Semi-spherical head tracking probe is shown in Figure 6.11. In simulating the probe, the local marker noise value for the probe was set at 0.127 mm to coincide with the manufacturing data for the accuracy of marker placement.

The simulation of this probe predicted an orientation error of  $0.54 \pm 0.24$  deg, shown in Figures 6.12 and 6.13. The black rectangle shown in Figure 6.12 indicates the area displayed in Figure 6.13. In translation, the model predicts a translation error of  $0.24 \pm 0.1$  mm. Experimentally, the accuracy in orientation and position was found to be  $0.60 \pm 0.03$  deg and  $0.225 \pm 0.05$  mm, respectively. We can see that there is a discrepancy between the precision of the predicted and experimental results in angle. Initially, the difference in the means of the angular predicted and experimental results was on the order of 40%. However, upon observing this, we increased the number of samples in the predictive model to 10,000 to obtain a better estimate of the mean. The remaining discrepancy between the predicted and experimental precision is likely compounded by two factors. First, the markers for this tracking probe are tightly packed, relative to the other

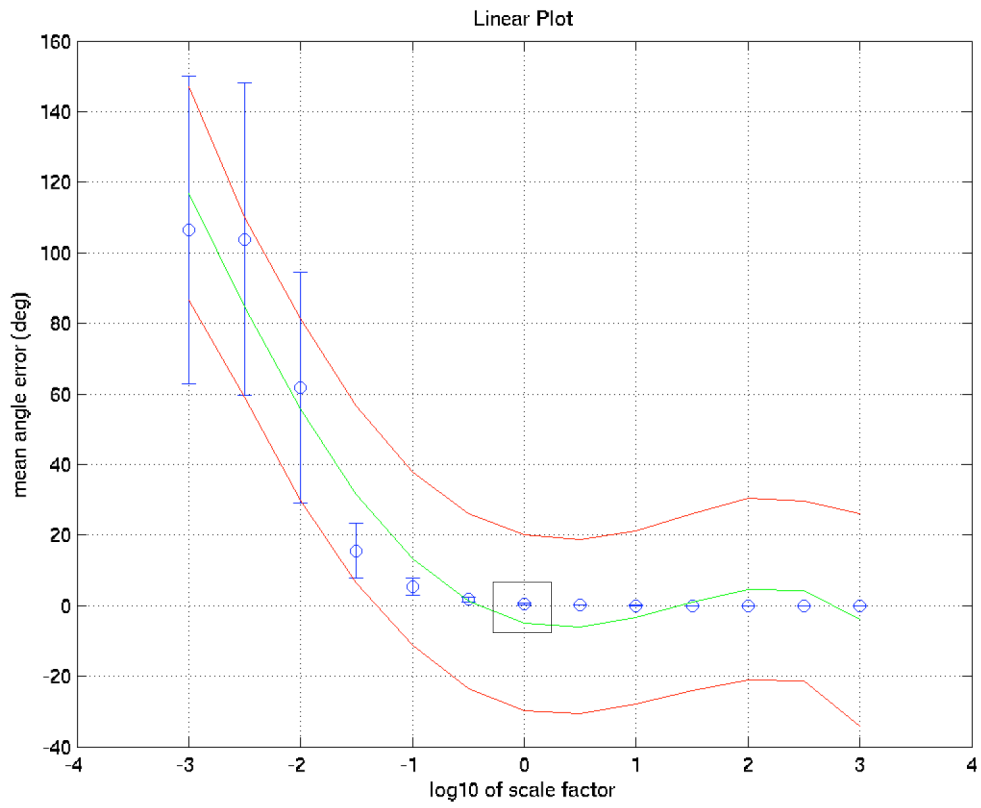


Figure 6.12: Simulation Results from Semi-spherical Head Tracking probe

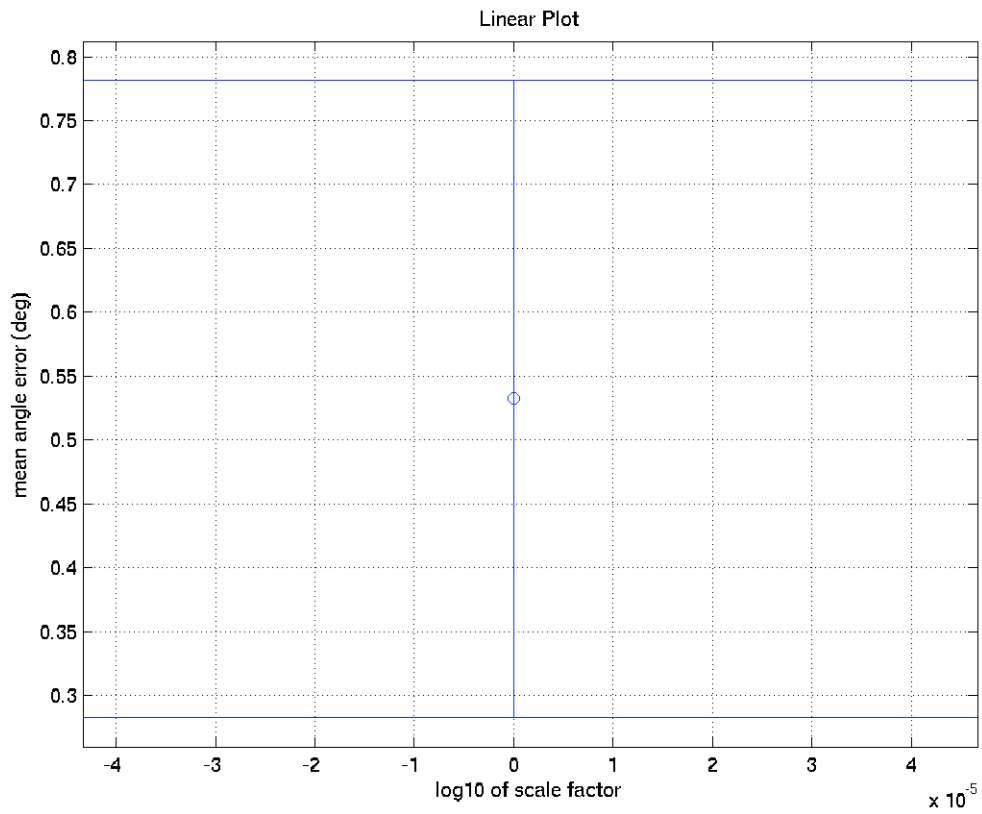


Figure 6.13: Simulation Results from Semi-spherical Head Tracking Probe (at close range)



probes examined. The other factor is that our model for prediction is based upon first-order error propagation. Because the markers are in close proximity with a relatively large amount of variance (due to the first-order error model), it is not surprising to observe this discrepancy.

### 6.3.2 A Conformally Mapped Probe – HMD

The geometry of an integrated head tracking probe was designed using a 3D model obtained from CAD data. Using the Viewpoints Algorithm, we placed 11 markers on the front surface of the HMD. The specific viewpoints were spread at 45 degree increments on a 90 degree solid-angle wedge of an intermediary sphere. The final mapping of the markers is shown in Figure 6.14

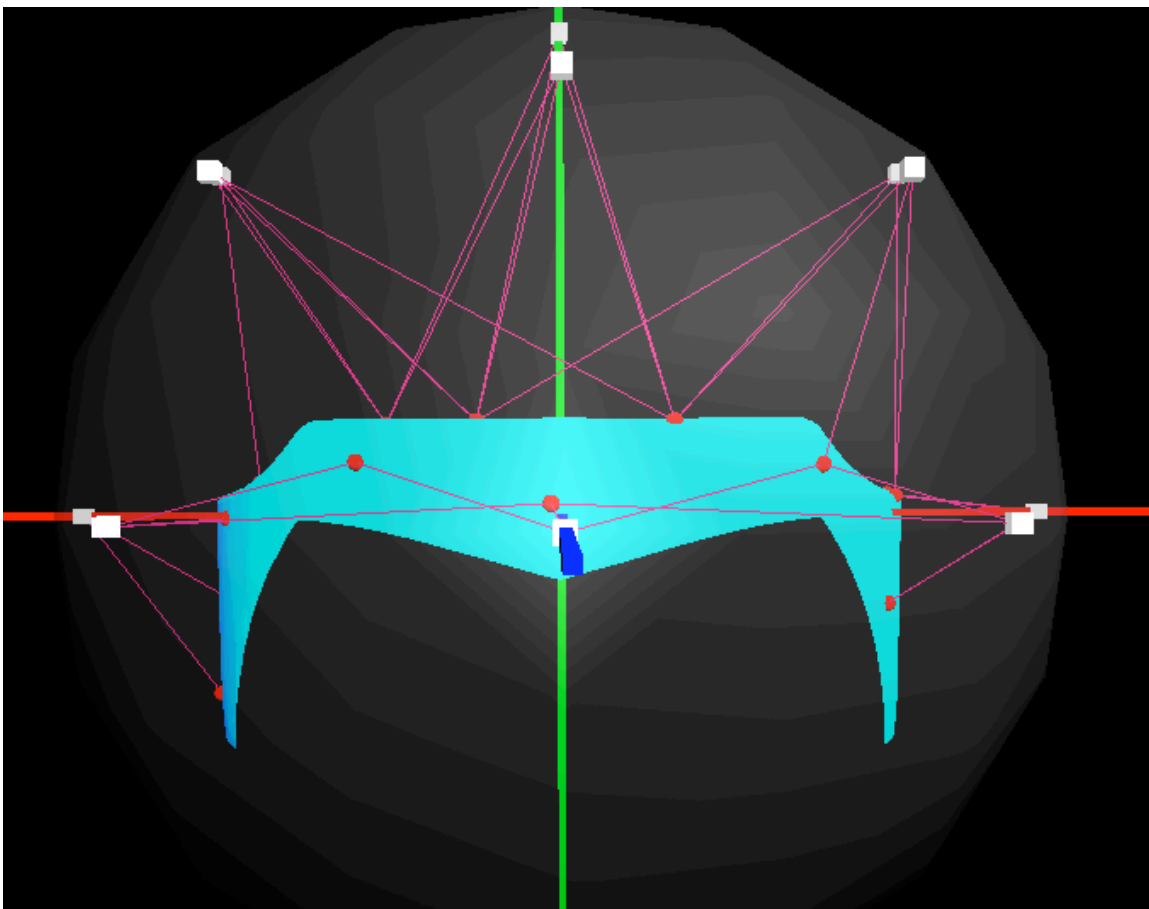


Figure 6.14: VRML Visualization of a Conformally Mapped Tracking Probe

The markers were then placed physically on the HMD using GeoMagic from Raindrop Software. In this package, the ability is present to determine the distance between specified points in the model. During the Viewpoint mapping technique, we chose the local coordinate system to be coincident with that of the 3D model. Therefore, when viewing the model in GeoMagic we were then able to determine the location of prominent features relative to the origin of the local coordinate system, then determine the distance from a given feature to a marker location. The procedure was done by hand using a digital caliper, so additional errors may be present. However, in the stereolithography process, the marker locations can be colored, so that the markers may be placed with high accuracy in future designs.



Figure 6.15: Realization of a Conformally Mapped Tracking Probe

The conformally mapped HMD tracking probe is shown in Figure 6.15. The local marker noise value for the probe was set at 0.1 mm to coincide with the stereolithography manufacturing data for the accuracy of marker placement. The simulation of this probe predicted an error of  $0.0325 \pm 0.02$  deg in orientation, shown in Figures 6.16 and 6.17, with an accuracy of  $0.14 \pm 0.08$  mm in translation. The black rectangle show in Figure 6.16 indicates the area displayed in Figure 6.17. The plots shown are simulating different global noise values for the HMD probe configuration. Experimentally, we obtained an accuracy of  $0.028 \pm 0.001$  degrees in orientation and  $0.11 \pm 0.01$  mm in position.

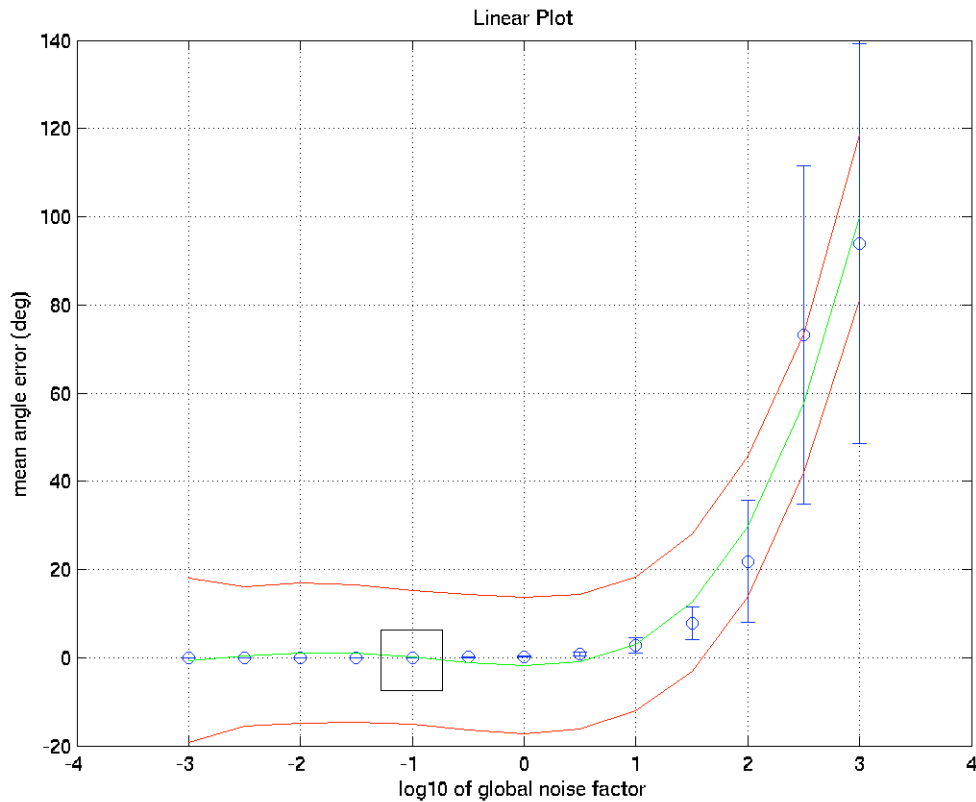


Figure 6.16: Simulation Results of the HMD Tracking Probe

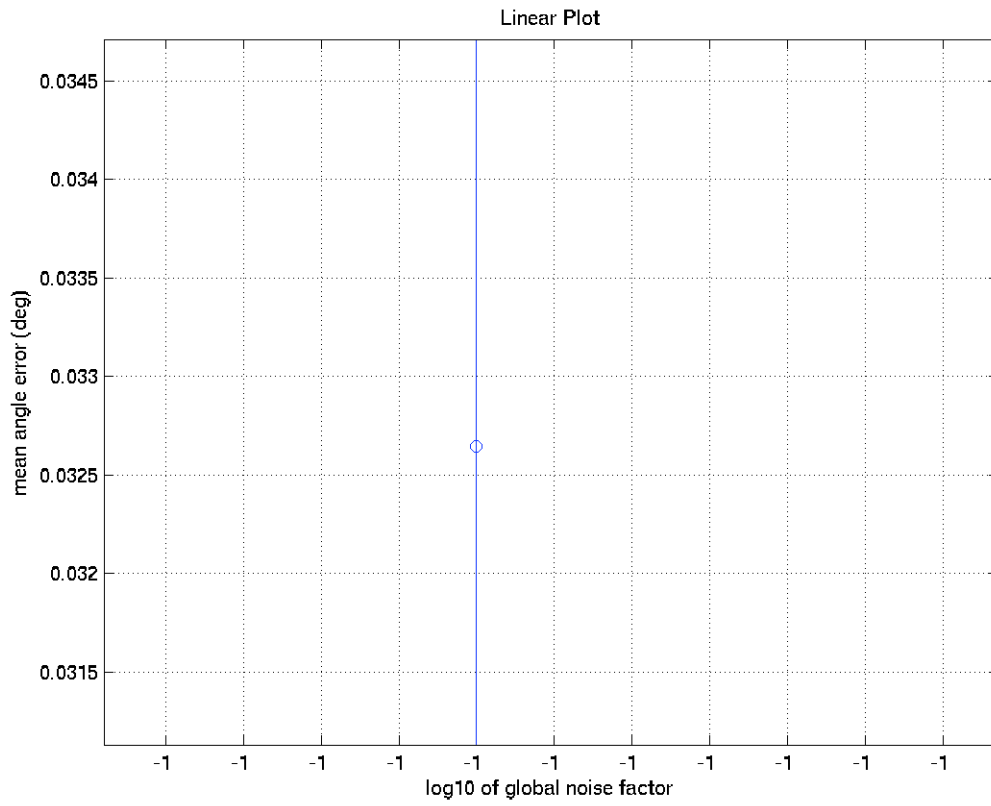


Figure 6.17: Simulation Results of the HMD Tracking Probe (at close range)

## CHAPTER 7: DISCUSSION AND CONCLUSIONS

The main contribution of this work is the creation of a mathematical model for predicting errors in pose estimation. We have quantified the effects of tracker and fabrication noise and the effects of probe size on pose errors. Furthermore, we have combined these results, introduced novel marker mapping techniques, and created a framework for the design of conformal tracking probes based upon environmental and application constraints. The tracking pose error results achieved for the conformal head tracking point to the promise of this approach for use AR environments. To achieve a registration error of 1 mm, an the pose error for a tracking probe must be less than 0.057 deg [Holloway, 1995]. With the design presented, we have obtained a static accuracy good enough to achieve this standard. We now discuss further some of the interesting problems that could provide directions for future research.

### 7.1 Further Insights for Conformal Probe Design

When mapping markers on an object, the number of markers used is not as important as how the markers are arranged. Figure 7.1 shows the results from a simulation where the number of markers placed on an object was increased dramatically. The probe simulated was a planar tracking probe and the markers were placed randomly. The results show that the pose error is reduced as the number of markers is increased. However, the quantity of markers added must be substantial to affect an increase. For example, for the probe in Figure 7.1, if the starting tracking probe was comprised of five markers, 20 markers would have to be added to gain a 50% increase in the accuracy of pose estimation. Moreover, adding 20 markers to the probe would only improve the pose error from 0.06 mm to 0.03 mm, which may not have a noticeable impact upon a particular application . Similar results were obtained when the type of tracking probe simulated was a sphere. However, further study will be required to model the effect empirically and to quantify the effect mathematically.

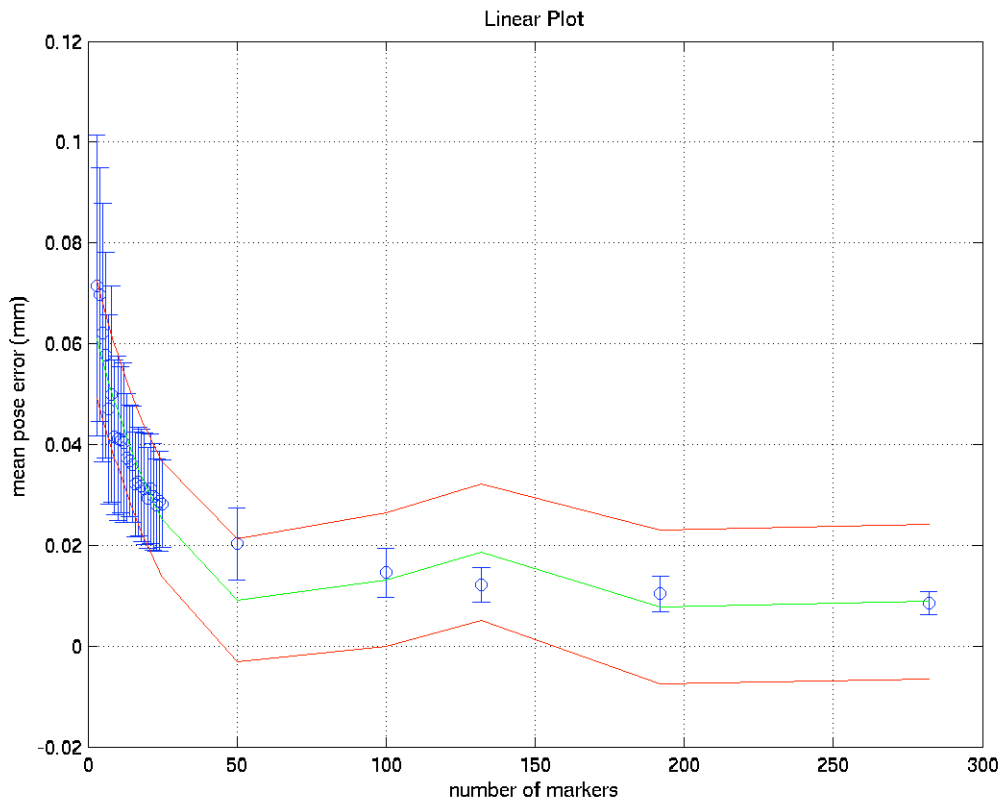


Figure 7.1: A Plot of Pose Error vs. large numbers of markers

## 7.2 Marker Mapping Concerns

The two mapping algorithms presented are both optimization-based algorithms. However, the mapping algorithm of choice is the Viewpoints Algorithm because it not only finds optimal positions for the markers, but it also minimizes the number of markers needed to fulfill the designer's field of regard constraint. The Viewpoints Algorithm is not foolproof, though, because of inconsistencies with the function which tests for whether a triangle is seen or not. In this test, the triangle normal is compared to the vector from the viewpoint being tested at the time. If the resulting dot product between the two vectors is negative, then the triangle can be "seen." Unfortunately, if one imagines a viewpoint on one side of the letter "M," there may be up to three triangles along a ray cast from the viewpoint through the "M" that may pass the test. This is a current problem in the field of computer graphics, and will be addressed in the future. Nevertheless, for relatively simple objects, the algorithm works well.

Both marker mapping algorithms presented rely upon having a 3D model of the object upon which the markers are to be mapped. However, within the process of mapping markers on an object, there are potential issues concerning the model itself. The problem lies in the triangle density of the 3-D model used to represent the real object. The triangle density is the number of triangles per unit of surface area. If a 3D model has a large number of small triangles, then the triangle density is high. As such, a high triangle density is desirable because it would be a more accurate representation of the object that will have markers placed upon it. Still, the Viewpoints Algorithm may end up mapping many markers in quite a small area. A solution to this issue is to apply a minimum distance constraint to the algorithm during the mapping process.

## 7.3 Numerical Limitations

In the simulations, there are places where the possibility of errors induced by numerical inaccuracies is increased. First, the pose error formulation uses matrices instead of quaternions to represent

rotations. Matrices require more numbers to represent an equivalent quaternion rotation, therefore matrix operations are more subject to numerical drift when used for comparable transformations. Second, the tracker noise may not be well-approximated by a Gaussian distribution. Third, in fitting a fourth-order polynomial to the simulation data, the correlation coefficient (which expresses the goodness of fit) ranged between 0.62 - 0.75 when examining data generated from independent variables in the range of  $\log(0.001)$  to  $\log(1000)$ . If we limit the independent variable to the range  $\log(0.01)$  to  $\log(100)$ , a superior exponential curve fit results. However, the dynamic range of the simulation is naturally limited. Finally, because we generate random distributions each time the simulation is run, we have variability in results. At small noise levels, the difference is negligible, but larger noise levels induce greater variability. Therefore, in the future, the tradeoff between static noise vectors and dynamically generated noise must be examined.

#### 7.4 Marker/Tracker Issues

If the intensity of the marker field of emission is non-uniform or its signal is somehow distorted, the accuracy in pose estimation may be affected. For example, in an optical tracking scenario with active markers, there may be distortions in the reported position of a marker based upon the angle at which it is detected. In an acoustic tracking system, there may be inaccuracies in reported marker positions due to ambient noise.

To address these issues, a practitioner should observe the behavior of a single marker within the tracking volume. With a precise rotation platform, the angle at which the tracker detects the marker at half power could be detected. From there, a functional relationship between SNR and marker angle could be extrapolated. Once this is done, knowing the position of a simulated probe, the appropriate noise value could be assigned to each marker on the probe.



## 7.5 Scale Factor in Pose Estimation

In the formulation presented, we assume unity scale factor because our investigation concerns rigid tracking probes. However, as a consequence of noise effects on the marker distribution, we may notice some errors due to an apparent scaling. We can examine the least squares equation, with the scale factor computed as done in [Horn, 1987]. A comprehensive analysis of the change in scale with noise will be needed to improve the performance of the model.

## 7.6 Future Work

The research presented addresses a critical need for quantification and assessment for marker based tracking in virtual environments. We have presented a framework for designing tracking probes, introduced a mathematical model for error propagation in pose estimation, and quantified the impact of environmental variables. As with most research, however, the process of finding answers often leads to more questions and directions for future efforts.

One such effort is an examination of the effects of tracker precision. Intuitively, having a tracker with good precision is better than a tracker with poor precision. Within our framework, we seek to incorporate these effects quantitatively. Through an examination of second order statistics, it should be possible to quantify precision effects. Based upon the results from Chapter 5, tracker precision problems are likely to affect the magnitude of tracker noise values, resulting in poorer pose estimations.

Also of importance is the need to understand the effects of changing the arrangement of markers on a tracking probe, non-uniformly, i.e., changing the probe shape. By isolating the individual effects of marker motion, a more superior probe design can be achieved. Section 5.2.2 proposed the foundation of a quantitative analysis of individual marker motion. Extending this foundation, we propose an investigation into the properties of the  $\mathbf{H}$  matrix. An analysis of variance on the elements of this matrix would be a starting point. Then, propagating the covariance effects to the

eigenvalues of  $A$  and  $B$  may provide insight into how to quantify individual marker effects. Each marker could then be weighted according to its importance to the pose estimation accuracy of the probe.

The issues concerned with marker weighting lead to a discussion of relative motion between markers on the tracking probe. In the case that the probe is not rigid, the markers may move with respect to each other. A way to account for relative motion is to weight each marker on the tracking probe according to its fidelity. In this way, more importance is given to the higher fidelity markers. However, the process of assigning weights to markers is not straightforward. One way to determine the weight for a marker is to move the probe through a predefined motion that approximates the movement to be tracked and determine the weighting based upon the standard deviation of marker motions relative to the centroid. A more robust method would be to apply an adaptive filtering algorithm to the positions of the markers on the non-rigid object. In this way, the weights could be updated dynamically and the properties of the probe deformation could be examined. The inclusion of such capability in this framework is highly desired.

Because tracking probes are meant for dynamic usage, a logical next step would be an extension of the framework to include dynamic errors. Adding dynamic capabilities to the current framework will require a significant effort. To start, for any marker based tracking scheme, there is an apparent deformation of the tracking probe due to the fact that all the marker positions are not acquired simultaneously. Next, the relationship between the speed of the probe movement and the update rate of the tracking system must be quantified. Then, the latency of tracker measurements must be studied to prevent large errors from appearing within the framework. Finally, any adaptive weighting applied for non-rigid tracking probes must be predictive. In fact, predictive algorithms may have to be used to overcome the tracker latency. There are a few large-scale research topics within this problem, making it quite appealing.

An implementation task to be performed is the migration of the pose error determination algorithm to C++. This would result in faster computation and greater freedom in implementation.

Users could import their own mathematical functions and modify the algorithms utilized, which are some of the benefits of modular code. The end users could also selectively utilize the portions of the framework most germane to their research, if desired. For our purposes, having the framework in C++ allows incorporation into other in-house VE utilities, such as the Distributed Artificial Reality Environment (DARE) Framework [Hamza-Lup et al., 2004].

A fast, native implementation of the conformal framework would lead to its ultimate incarnation, that is, a fully integrated, dynamic probe design procedure. Markers placed on the probe would be visualized as they are mapped. The pose estimation procedure would update dynamically as more markers were added. The markers on the probe could be moved through a GUI and the probe pose error would adjust dynamically along with the error contribution of the current marker.

The ideas and directions for the ongoing research underscore the potential for cross-pollination with other disciplines and research areas. For instance, a dynamic probe design system could be used for network assessment, with the markers representing nodes and the marker weights representing latencies from the current network position. Maximizing the field of regard of the probe would spread the computational or traffic load equally among the computing nodes. Another possibility is protein synthesis or microbiological research. The tracking probe could represent a gene of a certain shape and the markers proteins that occur within the gene. The weighting of the markers could be calculated in such a way as to ensure that the illegal protein sequences do not occur. By finding the optimal configuration of proteins, a new antibiotic could be formed or interesting genetic sequences studied. Indeed, a vast number of applications could be found for a dynamic, conformal tracking framework.

APPENDIX  
SOLUTION OF EIGENVALUE COMPONENTS

We can express  $a^2$  as

$$\begin{aligned}
a^2 \approx & \alpha^2 [ (s_{12}\sigma_{11})^2 + (s_{22}\sigma_{21})^2 + (s_{32}\sigma_{31})^2 + (s_{11}\sigma_{12})^2 + (s_{21}\sigma_{22})^2 + (s_{31}\sigma_{32})^2 \\
& + 2(s_{12}\sigma_{11})(s_{22}\sigma_{21}) + 2(s_{12}\sigma_{11})(s_{32}\sigma_{31}) + 2(s_{12}\sigma_{11})(s_{11}\sigma_{12}) \\
& + 2(s_{12}\sigma_{11})(s_{21}\sigma_{22}) + 2(s_{12}\sigma_{11})(s_{31}\sigma_{32}) + 2(s_{22}\sigma_{21})(s_{32}\sigma_{31}) \\
& + 2(s_{22}\sigma_{21})(s_{11}\sigma_{12}) + 2(s_{22}\sigma_{21})(s_{21}\sigma_{22}) + 2(s_{22}\sigma_{21})(s_{31}\sigma_{32}) \\
& + 2(s_{32}\sigma_{31})(s_{11}\sigma_{12}) + 2(s_{32}\sigma_{31})(s_{21}\sigma_{22}) + 2(s_{32}\sigma_{31})(s_{31}\sigma_{32}) \\
& + 2(s_{11}\sigma_{12})(s_{21}\sigma_{22}) + 2(s_{11}\sigma_{12})(s_{31}\sigma_{32}) + 2(s_{21}\sigma_{22})(s_{31}\sigma_{32}) ] . \tag{1}
\end{aligned}$$

Examining the mixed polynomial terms,

$$\begin{aligned}
& (s_{nm}\sigma_{pq})(s_{rt}\sigma_{uv}) = s_{nm}s_{rt}\sigma_{pq}\sigma_{uv} \tag{2} \\
= & \left[ \sum_{k=1}^K w_k \widetilde{x}_{kn} \widetilde{y}_{km} \right] \left[ \sum_{k=1}^K w_k \widetilde{x}_{kr} \widetilde{y}_{kt} \right] \left[ \sum_{k=1}^K (w_k \Delta \widetilde{x}_{kp} \widetilde{y}_{kq} + w_k \widetilde{x}_{kp} \Delta \widetilde{y}_{kq}) \right] \\
& \left[ \sum_{k=1}^K (w_k \Delta \widetilde{x}_{ku} \widetilde{y}_{kv} + w_k \widetilde{x}_{ku} \Delta \widetilde{y}_{kv}) \right] \tag{3}
\end{aligned}$$

$$\begin{aligned}
= & (w_1 \widetilde{x}_{1n} \widetilde{y}_{1m} + \dots + w_k \widetilde{x}_{kn} \widetilde{y}_{km}) (w_1 \widetilde{x}_{1r} \widetilde{y}_{1t} + \dots + w_k \widetilde{x}_{kr} \widetilde{y}_{kt}) \\
& (w_1 \Delta \widetilde{x}_{1p} \widetilde{y}_{1q} + w_1 \widetilde{x}_{1p} \Delta \widetilde{y}_{1q} + \dots + w_k \Delta \widetilde{x}_{kp} \widetilde{y}_{kq} + w_k \widetilde{x}_{kp} \Delta \widetilde{y}_{kq}) \\
& (w_1 \Delta \widetilde{x}_{1u} \widetilde{y}_{1v} + w_1 \widetilde{x}_{1u} \Delta \widetilde{y}_{1v} + \dots + w_k \Delta \widetilde{x}_{ku} \widetilde{y}_{kv} + w_k \widetilde{x}_{ku} \Delta \widetilde{y}_{kv}) \tag{4}
\end{aligned}$$

Assuming  $K = 2$ , we get

$$\begin{aligned}
= & (w_1 \widetilde{x}_{1n} \widetilde{y}_{1m} + w_2 \widetilde{x}_{2n} \widetilde{y}_{2m}) (w_1 \widetilde{x}_{1r} \widetilde{y}_{1t} + w_2 \widetilde{x}_{2r} \widetilde{y}_{2t}) \\
& (w_1 \Delta \widetilde{x}_{1p} \widetilde{y}_{1q} + w_1 \widetilde{x}_{1p} \Delta \widetilde{y}_{1q} + w_2 \Delta \widetilde{x}_{2p} \widetilde{y}_{2q} + w_2 \widetilde{x}_{2p} \Delta \widetilde{y}_{2q}) \\
& (w_1 \Delta \widetilde{x}_{1u} \widetilde{y}_{1v} + w_1 \widetilde{x}_{1u} \Delta \widetilde{y}_{1v} + w_2 \Delta \widetilde{x}_{2u} \widetilde{y}_{2v} + w_2 \widetilde{x}_{2u} \Delta \widetilde{y}_{2v}) \tag{5}
\end{aligned}$$

$$\begin{aligned}
&= (w_1^2 \widetilde{x}_{1n} \widetilde{y}_{1m} \widetilde{x}_{1r} \widetilde{y}_{1t} + w_2^2 \widetilde{x}_{2n} \widetilde{y}_{2m} \widetilde{x}_{2r} \widetilde{y}_{2t} + w_1 w_2 \widetilde{x}_{1n} \widetilde{y}_{1m} \widetilde{x}_{2n} \widetilde{y}_{2m} \\
&\quad + w_1 w_2 \widetilde{x}_{1r} \widetilde{y}_{1t} \widetilde{x}_{2r} \widetilde{y}_{2t}) \\
&\quad (w_1 \Delta \widetilde{x}_{1p} \widetilde{y}_{1q} + w_1 \widetilde{x}_{1p} \Delta \widetilde{y}_{1q} + w_2 \Delta \widetilde{x}_{2p} \widetilde{y}_{2q} + w_2 \widetilde{x}_{2p} \Delta \widetilde{y}_{2q}) \\
&\quad (w_1 \Delta \widetilde{x}_{1u} \widetilde{y}_{1v} + w_1 \widetilde{x}_{1u} \Delta \widetilde{y}_{1v} + w_2 \Delta \widetilde{x}_{2u} \widetilde{y}_{2v} + w_2 \widetilde{x}_{2u} \Delta \widetilde{y}_{2v}) \tag{6}
\end{aligned}$$

What becomes apparent from the  $\sigma_{pq}\sigma_{uv}$  multiplication is a  $\Delta x_k^2$  or  $\Delta y_k^2$  term will result only if  $p = u$  or  $q = v$ , respectively. Since the mixed  $\Delta x_k \Delta y_k$  terms go to zero when looking at an ensemble average, the expression becomes

$$\begin{aligned}
s_{nm} s_{rt} \sigma_{pq} \sigma_{uv} &= (w_1^2 \widetilde{x}_{1n} \widetilde{y}_{1m} \widetilde{x}_{1r} \widetilde{y}_{1t} + w_2^2 \widetilde{x}_{2n} \widetilde{y}_{2m} \widetilde{x}_{2r} \widetilde{y}_{2t} + w_1 w_2 \widetilde{x}_{1n} \widetilde{y}_{1m} \widetilde{x}_{2n} \widetilde{y}_{2m} \\
&\quad + w_1 w_2 \widetilde{x}_{1r} \widetilde{y}_{1t} \widetilde{x}_{2r} \widetilde{y}_{2t}) \\
&\quad (w_1^2 \Delta \widetilde{x}_{1p} \widetilde{y}_{1q} \Delta \widetilde{x}_{1u} \widetilde{y}_{1v} + w_1^2 \widetilde{x}_{1p} \Delta \widetilde{y}_{1q} \widetilde{x}_{1u} \Delta \widetilde{y}_{1v} \\
&\quad + w_2^2 \Delta \widetilde{x}_{2p} \widetilde{y}_{2q} \Delta \widetilde{x}_{2u} \widetilde{y}_{2v} \\
&\quad + w_2^2 \widetilde{x}_{2p} \Delta \widetilde{y}_{2q} \widetilde{x}_{2u} \Delta \widetilde{y}_{2v}) \tag{7}
\end{aligned}$$

## LIST OF REFERENCES

- [3rdTech Inc., 2002] 3rdTech Inc. (2002). Hi ball 3000 wide area tracker and 3D digitizer (data sheet). Chapel Hill, NC.
- [Allen and Welch, 2004] Allen, B. and Welch, G. (2004). Artemis: A system for the analysis and comparison of the expected performance for tracking and motion capture systems. In *Proceedings of SIGGRAPH '04*. ACM SIGGRAPH (in Press).
- [Arun et al., 1987] Arun, K., Huang, T., and Blostein, S. (1987). Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(5):698–700.
- [Ascension Corp., 1990] Ascension Corp. (1990). Flock of birds manual. Burlington, VT.
- [Ascension Corp., 2001] Ascension Corp. (2001). Motion star wireless: Installation and operation guide. Burlington, VT.
- [Ascension Corp., 2002] Ascension Corp. (2002). laserBIRD installation and operation guide. Burlington, VT.
- [Azuma and Ward, 1991] Azuma, R. and Ward, M. (1991). Space-resection by collinearity: Mathematics behind the optical ceiling head-tracker. Technical Report TR 91-048, UNC Chapel Hill.
- [Bajura et al., 1992] Bajura, M., Fuchs, H., and Ohbuchi, R. (1992). Merging virtual objects with the real world: Seeing ultrasound imagery within the patient. In *Proceedings of SIGGRAPH '92*, pages 203–210. ACM SIGGRAPH.
- [Brooks et al., 1990] Brooks, F., Ouh-Young, M., Batter, J., and Kilpatrick, P. (1990). Project GROPE - haptics displays for scientific visualization. *Computer Graphics*, 24(4):177–185.
- [Burdea and Coiffet, 1994] Burdea, G. and Coiffet, P. (1994). *Virtual Reality Technology*. Wiley Interscience.

- [Caudell and Mizell, 1992] Caudell, T. and Mizell, D. (1992). Augmented reality: An application of heads-up display technology to manual manufacturing processes. In *Proceedings of Hawaii International Conference on System Sciences*, pages 659–669.
- [Cruz-Neira et al., 1993] Cruz-Neira, C., Sandin, D., and DeFanti, T. (1993). Surround-screen projection-based virtual reality: the design and implementation of the cave. In *Proceedings of SIGGRAPH '93*, pages 135–142. ACM SIGGRAPH.
- [Czernuszenko et al., 1997] Czernuszenko, M., Pape, D., Sandin, D., DeFanti, T., Dawe, G., and Brown, M. (1997). The immersadesk and infinity wall projection-based virtual reality displays. *Computer Graphics*, 31(2):46–49.
- [Davis et al., 2002] Davis, L., Rolland, J., Parsons, R., and Clarkson, E. (2002). Methods for designing head-tracking probes. In *Proceedings of the Joint Conference on Information Sciences (JCIS)*, pages 498–502.
- [de Ridder, 2004] de Ridder, D. (2004). Principal component analysis. In Fisher, R., editor, *CVOnline: Online Compendium of Computer Vision*. <http://www.ph.tn.tudelft.nl/dick/cvonline/pca/pca.html>.
- [Dowland, 1993] Dowland, K. (1993). Simulated annealing. In Reeves, C., editor, *Modern Heuristic Techniques for Combinatorial Problems*. Wiley & Sons.
- [Durlach and Mavor, 1994] Durlach, N. and Mavor, A. (1994). *Virtual Reality: Scientific and Technical Challenges*. National Academy Press.
- [Edwards et al., 2000] Edwards, P., King, A., Maurer, C., de Cunha, D., Hawkes, D., Hill, D., Gaston, R., Fenlon, M., Juszczak, A., Strong, A., Chandler, C., and Gleeson, M. (2000). Design and evaluation of a system for microscope-assisted guided interventions (MAGI). *IEEE Transactions on Medical Imaging*, 19(11):1082–1093.



- [Farrell and Barth, 1999] Farrell, J. and Barth, M. (1999). *The Global Positioning System and Inertial Navigation*. McGraw-Hill.
- [Feiner et al., 1993] Feiner, S., MacIntyre, B., and Seligmann, D. (1993). Knowledge-based augmented reality. *Communications of the ACM*, 36(7):52–62.
- [Ferrin, 1991] Ferrin, F. (1991). Survey of helmet tracking technologies. *Large-Screen-Projection, Avionic, and Helmet-Mounted Displays*, 1456:86–94.
- [Fisher et al., 1986] Fisher, S., McGreevy, M., Humphries, J., and Robinett, W. (1986). Virtual environment display system. In *Proceedings of the Workshop on Interactive 3D Graphics*, pages 77–87. ACM.
- [Foxlin and Durlach, 1994] Foxlin, E. and Durlach, N. (1994). An inertial head-orientation tracker with automatic drift compensation for use with hmd's. In Singh, G., Feiner, S., and Thalmann, D., editors, *Proc. Virtual Reality Software & Technology 94*, pages 159–174. World Scientific.
- [Foxlin et al., 1998] Foxlin, E., Harrington, M., and Pfeifer, G. (1998). Constellation: A wide-range wireless motion-tracking system for augmented reality and virtual set applications. In *Proceedings of SIGGRAPH '98*, pages 371–378. ACM SIGGRAPH.
- [Hamza-Lup et al., 2004] Hamza-Lup, F., Davis, L., Fuhrman, C., Hughes, C., and Rolland, J. (2004). The distributed artificial reality environment (DARE) framework. Technical Report TR-04-001, University of Central Florida.
- [Hamza-Lup et al., 2002] Hamza-Lup, F., Davis, L., Hughes, C., and Rolland, J. (2002). Marker mapping techniques for augmented reality. In *Proceedings of the International Symposium on Computer and Information Sciences (ISCIS 2002)*, pages 152–156.
- [Haralick et al., 1989] Haralick, R. M., Joo, H., Lee, C., Zhuang, X., Vaidya, V., and Kim, M. (1989). Pose estimation from corresponding point data. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(6):1426–1446.

- [Holloway, 1995] Holloway, R. (1995). *Registration Errors in Augmented Reality Systems*. PhD thesis, University of North Carolina - Chapel Hill.
- [Horn, 1987] Horn, B. (1987). Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642.
- [Hotelling, 1933] Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441, 498–520.
- [Huber, 1981] Huber, P. (1981). *Robust Statistics*. Wiley Series in Probability and Mathematical Statistics. Wiley Interscience.
- [Intersense, Inc., 2000] Intersense, Inc. (2000). User manual for IS-900 VET and VWT. Burlington, MA.
- [Karhunen, 1947] Karhunen, K. (1947). Über lineare methoden in der wahrscheinlichkeitsrechnung. *Annales Academiae Fennicae, Series A*, 37:3–79. (Translation by Selin, I. in "On Linear Methods in Probability Theory", RAND Corporation, Santa Monica, California, Rep. T-131, Aug. 1960) .
- [Kato and Billinghurst, 1999] Kato, H. and Billinghurst, M. (1999). Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *The 2nd International Workshop on Augmented Reality*, pages 85–94.
- [Kirkpatrick et al., 1983] Kirkpatrick, S., Gelatt, C., and Vecchi, M. (1983). Optimization by simulated annealing. *Science*, 220:671–680.
- [Krueger, 1977] Krueger, M. (1977). Responsive environments. In *Proceedings of the National Computer Conference*, pages 423–433.
- [Krueger, 1985] Krueger, M. (1985). Videoplace - an artificial reality. In *Proceedings of SIGCHI '85*, pages 35–40. ACM.

- [Lippman, 1980] Lippman, A. (1980). Movie maps: An application of optical video disks to computer graphics. *Computer Graphics*, 14(3):32–42.
- [Loève, M, 1948] Loève, M (1948). Fonctions aléatoires de seconde ordre. In Lévy, P., editor, *Processus Stochastiques et Mouvement Brownien*. Hermann, Paris.
- [Logitech Inc., 1992] Logitech Inc. (1992). 3d mouse and head tracker technical reference manual.
- [Mellor, 1995] Mellor, J. (1995). Realtime cameral calibration for enhanced reality visualization. In *Proceedings of the First International Conference on Computer Vision*. Virtual Reality and Robotics in Medicine.
- [Metropolis et al., 1953] Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E. (1953). Equation of state calculation by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092.
- [Meyer et al., 1992] Meyer, K., Applewhite, H., and Biocca, F. (1992). A survey of position trackers. *Presence: Teleoperators and Virtual Environments*, 1(2):173–200.
- [Milgram and Kishino, 1994] Milgram, P. and Kishino, F. (1994). A taxonomy of mixed reality visual displays. *IECE Trans. on Information and Systems—Special Issue on Networked Reality*, E77-D(12):1321–1329.
- [Morris and Donath, 1993] Morris, T. and Donath, M. (1993). Using a maximum error statistic to evaluate measurement errors in 3d position and orientation tracking systems. *Presence: Teleoperators and Virtual Environments*, 2(4):314–343.
- [Neumann et al., 1999] Neumann, U., You, S., Cho, Y., Lee, J., and Park, J. (1999). Augmented reality tracking in natural environments. In *International Symposium on Mixed Realities*.

- [Northern Digital Inc., 1999] Northern Digital Inc. (1999). OPTOTRAK getting started guide. Waterloo, ON.
- [Park et al., 1999] Park, J., Jiang, B., and Neumann, U. (1999). Vision-based pose computation: Robust and accurate augmented reality tracking. In *IEEE International Workshop on Augmented Reality*, pages 3–12.
- [Pearson, 1901] Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *The London, Edinburgh and Dublin Philosophical Magazine and Journal of Science*.
- [Polhemus Inc., 2000] Polhemus Inc. (2000). Star\*track description and preliminary system specification.
- [Raab et al., 1979] Raab, F., Blood, E., Steiner, O., and Jones, H. (1979). Magnetic position and orientation tracking system. *IEEE Transactions on Aerospace and Electronics Systems*, 15(5):709–717.
- [Rabinovich, 1995] Rabinovich, S. (1995). *Measurement Errors: Theory and Practice*. American Institute of Physics, New York.
- [Ravela et al., 1995] Ravela, S., Draper, B., Lim, J., and Weiss, R. (1995). Adaptive tracking and model registration across distinct aspects. In *IROS95*.
- [Rheingold, 1991] Rheingold, H. (1991). *Virtual Reality*. Simon and Shuster.
- [Rolland et al., 2000] Rolland, J., Davis, L., and Baillet, Y. (2000). A survey of tracking technology for virtual environments. In *Augmented Reality and Wearable Computers*, chapter 3. Erblaum.
- [Rolland and Fuchs, 2000] Rolland, J. and Fuchs, H. (2000). Optical versus video see-through head-mounted displays in medical visualization. *Presence: Teleoperators and Virtual Environments*, 9(3):287–309.

- [Sayood, 1996] Sayood, K. (1996). *Introduction to Data Compression*. Morgan Kaufmann Publishers, San Francisco.
- [Sobolev, 1962] Sobolev, S. (1962). The formulas of mechanical quadrature on the surface of a sphere. *Sibersk. Mat. Ž.*, 3:769–796.
- [Spoor and Veldpaus, 1980] Spoor, C. and Veldpaus, F. (1980). Technical note: Rigid body motion calculated from spatial coordinates of markers. *Journal of Biomechanics*, 13:391–393.
- [State et al., 1996] State, A., Hirota, G., Chen, D., Garrett, W., and Livingston, M. (1996). Superior augmented-reality registration by integrating landmark tracking and magnetic tracking. In *Proceedings of SIGGRAPH '96*, pages 429–438. ACM SIGGRAPH.
- [Sutherland, 1968] Sutherland, I. (1968). A head-mounted three-dimensional display. *AFIPS Conference Proceedings*, 33(Part I):757–764.
- [Tsai, 1987] Tsai, R. (1987). A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344.
- [Uenohara and Kanade, 1995] Uenohara, M. and Kanade, T. (1995). Vision-based object registration for real-time image overlay. *International Journal of Computers in Biology and Medicine*, 25(2):249–260.
- [Vogt et al., 2002] Vogt, S., Khamene, A., Sauer, F., and Niemann, H. (2002). Single camera tracking of marker clusters: Multiparameter cluster optimization and experimental verification. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR '02)*, pages 127–136. IEEE and ACM.
- [Wang et al., 1990] Wang, J., Chi, V., and Fuchs, H. (1990). A real-time optical 3d tracker for head-mounted display systems. *Computer Graphics*, 24:205–215.

- [Ward et al., 1992] Ward, M., Azuma, R., Bennet, R., Gottschalk, S., and Fuchs., H. (1992). A demonstrated optical tracker with scalable work area for head-mounted display systems. In *Proceedings of the Symposium on Interactive 3D Graphics*, pages 43–52. ACM.
- [Welch and Bishop, 1997] Welch, G. and Bishop, G. (1997). Scaat: Incremental tracking with incomplete information. In *Proceedings of SIGGRAPH '97*. ACM SIGGRAPH.
- [Welch and Foxlin, 2002] Welch, G. and Foxlin, E. (2002). Motion tracking: No silver bullet, but a respectable arsenal. *IEEE Computer Graphics and Applications, special issue on Tracking*, 22(6):24–38.
- [Woltring et al., 1985] Woltring, H., Huiskes, R., De Lange, A., and Veldpaus, F. (1985). Finite centroid and helical axis estimation from noisy landmark measurements in the study of human joint kinematics. *Journal of Biomechanics*, 18(5):379–389.
- [Zhang and Navab, 2000] Zhang, X. and Navab, N. (2000). Tracking and pose estimation for computer assisted localization in industrial environments. In *Proceedings of the International Symposium on Augmented Reality*. IEEE.
- [Zimmerman and Lanier, 1987] Zimmerman, T. and Lanier, J. (1987). Hand gesture interface device. In *Proceedings of CHI '87*, pages 189–192. ACM SIGCHI.