

This is to certify that the
thesis entitled

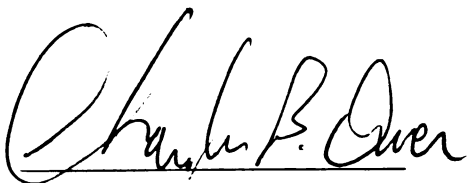
Evaluation of Calibration for Optical
See-Through Augmented Reality Systems

presented by

Erin Scott McGarrity

has been accepted towards fulfillment
of the requirements for

Master's degree in Computer Science
& Engineering



Major professor

Date 5-9-01

LIBRARY
Michigan State
University

PLACE IN RETURN BOX to remove this checkout from your record.
TO AVOID FINES return on or before date due.
MAY BE RECALLED with earlier due date if requested.

DATE DUE	DATE DUE	DATE DUE

EVALUATION OF CALIBRATION FOR OPTICAL
SEE-THROUGH AUGMENTED REALITY SYSTEMS

By

Erin Scott McGarrity

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Department of Computer Science and Engineering

2001

ABSTRACT

EVALUATION OF CALIBRATION FOR OPTICAL SEE-THROUGH
AUGMENTED REALITY SYSTEMS

By

Erin Scott McGarrity

This thesis describes a method for the metric evaluation of the registration accuracy of optical see-through head-mounted displays. This method provides a quantitative estimate of the calibration error while the user is *in situ*. This method was implemented as a module which can be added to an existing AR system. The module works by superimposing a virtual object, *e.g.* a small sphere, onto a known position in the workspace and asking the user to indicate where they perceive object to be located. The module then moves the object to another location and the process repeats. Finally, after a sufficient number of points have been displayed and their positions indicated, the module computes and displays the system errors by comparing the differences between the perceived and actual locations of the points.

Detailed descriptions of two optical see-through augmented reality systems, along with their calibration requirements, are described. The first system uses magnetic tracking while the second uses infrared video-based tracking. Both systems have workspace volumes of approximately $1.0m \times 1.0m \times 1.0m$. The results of the application of the method to both systems are shown. These results give a picture of the overall system error. They also show that the the observed error in an AR system is highly user dependent.

© Copyright May 9, 2001 by
Erin Scott McGarrity
All Rights Reserved

ACKNOWLEDGMENTS

To begin, I would like to thank the people who worked with me on this project. First, there's Dr. Charles Owen. He is a top notch advisor and editor. His thoughts and ideas about AR gave me a new way of looking at it. Next, I want to thank Dr. Mihran Tuceryan. It was Mihran who got me into AR in the first place. He also pointed me in the directions of many excellent restaurants and provided plenty to talk about outside the realm of AR. Next, I give thanks to Drs. George Stockman and Frank Biocca. Their inputs made me think more about the human component of AR. Also, thanks to Yakup, Nassir, Benedicte, Frank, Xhang and Ali for the lively discussions and brainstorming sessions at Siemens.

I would also like to thank four people who made indirect contributions and made my life much easier while I was working on my thesis. Let's start with Dr. Anil Jain. He made the financial stuff happen with SCR. There's also Ruth at Siemens who helped me find all those obscure references. Dr. Charles MacCluer helped to hone my math skills along the way. He also gave me lots of good advice about careers in academia. Finally, thanks to my wife, and best friend, Christine, for all her patience and support. She made the other parts of life (bills, car registrations, etc) go along smoothly.

TABLE OF CONTENTS

LIST OF TABLES	vi
LIST OF FIGURES	vii
1 Introduction	1
2 AR System Description	7
2.1 Magnetically Tracked System	7
2.2 Video Tracked System	9
3 Workspace Alignment	12
3.1 Rigid Alignment	13
3.2 Affine Alignment	16
3.3 Robust Alignment Computation	17
3.4 Experimental Results	22
4 HMD Calibration	30
4.1 Camera Model	31
4.2 Estimation of the Display Parameters	36
4.3 Calibration Procedure	37
5 Evaluating a Calibration	39
5.1 Evaluation Algorithm	41
5.2 Planar Calibration Validation	41
5.3 Experimental Results for Planar Verification	44
5.4 Analysis of Structured Errors	49
5.5 Conclusions	51
6 Summary and Conclusions	52
APPENDICES	54
A User Error Field Plots	55
BIBLIOGRAPHY	64

LIST OF TABLES

3.1	Pointwise standard deviation measurements.	18
3.2	Magnetic field distortion information.	19
3.3	Error analysis results.	24
5.1	Results for a single user from multiple viewing zones.	47
5.2	Average errors for different users from multiple viewing zones.	47
5.3	Maximum errors for different users from multiple viewing zones.	48

LIST OF FIGURES

1.1	Optical see-through display. The real scene is mixed with graphics using a 1/2-silvered mirror.	2
1.2	Coordinate systems for a magnetically-tracked AR system.	4
2.1	A photograph of the magnetically tracked system.	8
2.2	Magnetically tracked AR system.	8
2.3	AR system with infrared video-based tracking.	10
3.1	Pairwise inter-point differences for measured points.	20
3.2	Distribution of inter-point distances is approximately normal.	21
3.3	Errors using the affine methods with control calibration data.	26
3.4	Errors using the affine methods with noisy calibration data. The outliers are highly visible in the robust procedure.	27
3.5	Errors using the rigid methods with control calibration data.	28
3.6	Errors using the rigid methods with noisy calibration data.	29
4.1	The geometry of a simple pinhole camera model for perspective projection.	31
4.2	Coordinate transformations used for display calibration.	34
5.1	The optical see-through evaluation system.	42
5.2	Error is measured in the plane of the evaluation board as seen from the eyepoint, E . The true location is $\mathbf{p}_t$, the observed location is $\mathbf{p}_o$ and the projection of the observed point onto the board is $\mathbf{p}_b$	42
5.3	Error in the plane of the evaluation board as seen from the eye-points, E and E' . The error $\ \mathbf{p}_t - \mathbf{p}_b\ $ is larger than $\ \mathbf{p}_t - \mathbf{p}'_b\ $	43
5.4	Error fields for an AR system which uses a magnetic tracker. These are the planar errors as observed from three zones in the workspace.	45
5.5	Viewing zones for an evaluation.	46
5.6	Error fields for an AR system which uses a infrared video tracking system. $P0$ indicates the calibration origin and $T0$ indicates the tracker origin.	48
5.7	Reconstructed locations	49
A.1	Three-view errors from a user.	56
A.2	Three-view errors from a user.	57
A.3	Three-view errors from a user.	58
A.4	Three-view errors from a user.	59
A.5	Three-view errors from a user.	60
A.6	Three-view errors from a user.	61
A.7	Three-view errors from a user.	62
A.8	Three-view errors from a user. The outliers in this figure are from the user moving outside the range of the tracker.	63

Chapter 1

Introduction

Augmented Reality (AR) is the process of superimposing computer graphics onto the real world. The intent of AR is to enhance the amount of information about a place or object while it is being observed and interacted with. AR differs from Virtual Reality because it is not completely immersive. Instead, it is a composite of the real and the virtual. This paper provides a discussion about a particular branch of AR which is known as *optical see-through AR*. This branch uses a see-through head-mounted display (HMD) which mixes graphics with the scene using a 1/2-silvered mirror. Figure 1.1 shows a schematic for an optical see-through display.

There are many potential applications for AR. These include training, inspection and repair tasks. Examples include a circuit board inspection system [24] or a laser printer repair application [8]. It might also be possible to use an AR system in a telepresence application [6, 18]. In this application, a technician at a remote location can be guided through a repair process by an expert in the home office who is able to see what the technician sees using AR system. The expert can then provide

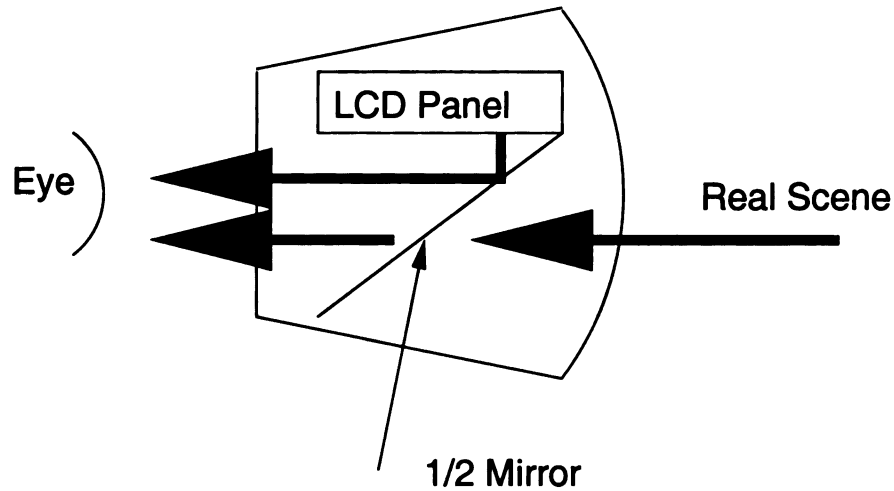


Figure 1.1: Optical see-through display. The real scene is mixed with graphics using a 1/2-silvered mirror.

multimedia feedback through the technician's display.

On another front, AR can be used for entertainment. There have been several applications in this vein such as “AR² Hockey”, “AquaGauntlet”, and “Welbo” [31]. In the same vein there is the potential for installing AR systems in museums and historic sites. Imagine going to the courthouse in Boston and seeing an augmented re-creation of the “Taxation Without Representation” or going to Dover Castle and seeing William I and his army crossing the English Channel and landing on the shores.

Another ideal application for AR is in computer-assisted surgery. AR can provide a form of X-ray eyes, allowing the surgeon to see through tissue and allowing results from instrumentation such as PET and MRI to be projected directly onto affected tissue. If a surgeon could see a virtual overlay of a tumor in a patient, they would be able to minimize the amount of invasion to the patient. A prototype of an AR surgical microscope has been built to this end [5]. There is much work which needs

to be performed, however, before such a system comes into common use in the operating room. Further exploration into tracking and registration methods needs to be performed. The validation of such a system is also crucial since there is no room for error in such an application.

In order for an augmented reality system to be effective, the virtual objects must be accurately positioned relative to the real objects. This implies that certain measurements or *calibrations* need to be made. These calibrations determine the transformations between the various components such as trackers, cameras and head-mounted displays (HMDs). The calibration requirements and the difficulties with their implementation depend on the architecture of the particular system.

Figure 1.2 shows the transformations between the objects in a typical magnetically-tracked AR setup. The objective of calibration is to estimate the transformations between each of these coordinate systems and a central reference coordinate system known as the *Workspace Coordinate System* (WCS). The WCS is generally at a fixed location relative to the operating environment. During the operation of a system, all of its components are referred to in the workspace coordinate system. The *Tracker Coordinate System* (TCS) is the coordinate system in which all measurements occur. The transformation between the TCS and WCS (W in the diagram) must be estimated since the origin of the TCS lies at some unknown position within the transmitter antenna. The *Marker Coordinate System* (MCS) is used to locate the user's head at runtime. The transform from the TCS into the MCS (M from the diagram) is measured in real time by the tracking system. The *Display Coordinate System* (DCS) is the coordinate system that the graphics are rendered in. The trans-

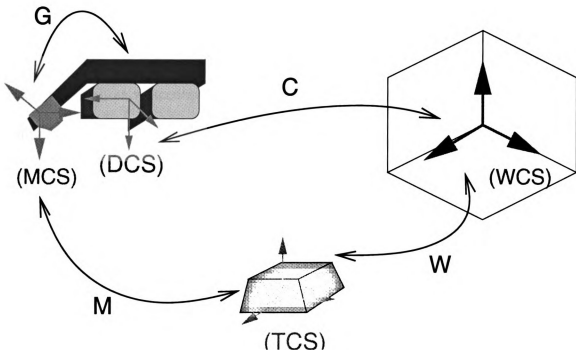


Figure 1.2: Coordinate systems for a magnetically-tracked AR system.

form between the MCS and the DCS (G from the diagram) is estimated during HMD calibration.

This paper is primarily concerned with HMD calibration and validation. HMD calibration is the practice of estimating the projective transformation which must be applied to the virtual objects in order to display them for the user. If the HMD is calibrated incorrectly, the virtual objects will not align properly with the real objects in the workspace. In principle HMD calibration is similar to camera calibration, yet, in practice there are many difficulties related to the tight coupling between the intrinsic and extrinsic parameters [11, 27].

A particular problem associated with calibrating optical see-through HMDs is determining the accuracy of the system. Traditional image-based methods are of little use since there is no way to access the augmented image which is formed on the

user's retina. Thus far, there have been two main approaches to accuracy verification for see-through displays. The first is to have the user calibrate the HMD and then report the qualitative accuracy of the alignment of a computer model to its real-world counterpart [33], *i.e.*, "acceptable," or "unacceptable." The second method uses a camera to replace the human eye and conducts image-based measurements to assess the accuracy of the alignment [27]. The former method is quick and easy, but it is subjective and gives no metric information. The latter method is performed by mounting a camera to the HMD so that it "sees" through the display panel. This is not very practical, nor does it guarantee that the measured accuracy will be observed by the user since a camera is only an approximation of the eye. Furthermore, the method is very tedious and must be performed offline since the camera is mounted where the user's head would normally be.

There are several general works which describe AR as a whole. Azuma classifies AR systems based on display types and levels of immersion [1]. Feiner, *et al.*, and Kutulakos and Vallino discuss workspace object representations in AR systems [8,21]. Klinker, *et al.*, describe the merging of graphics and computer vision [20]. Fuchs and Rolland compare and contrast optical and video see-through HMDs [10].

AR systems can be broadly categorized by the type of display they use. The first class is known as video see-through. Video see-through systems use a camera and display as an intermediary between the user and the scene. The video stream is captured by a camera, augmented and passed on to the user via a HMD or monitor. Much work has been done in this area. Tuceryan, *et al.*, describe the calibration requirements for a video based AR system [32]. Janin, Mizell and Caudell developed a

monocular system for assembling aircraft wiring harnesses [17]. Kato and Billingham describe a combined camera/HMD calibration scheme for tracking and augmenting fiducial markers in a scene [18]. Neumann and Cho present a method for improving the registration of such a system [29]. Mellor describes a procedure to calibrate a camera with embedded intrinsic parameters [28]. This method is very similar to the HMD calibration procedure described later in this thesis. Genc, *et al.*, discuss the validation of a stereo video see-through system [12]. Other researchers have been concentrating on registration of 3D medical data to surgical patients [3, 4, 13, 22, 34].

The second class of AR systems are called optical see-through systems. Tuceryan and Navab present a user-friendly calibration method for optical see-through HMDs [33]. Oishi and Tachi discuss the calibration of a rigidly mounted optical see-through display [30]. Azuma and Holloway analyze sources of registration error in AR systems [2, 14, 15]. This paper takes a similar direction, but also provides an online method for computing registration error. Previous work by this author in the realm of optical see-through calibration and validation can be found [11, 24–27].

The remainder of this paper is divided into four chapters. Chapter 2 describes the AR systems used for testing the verification method. Chapter 3 gives the details on the methods used to calibrate magnetic tracking systems. Chapter 4 describes the HMD calibration procedure used in both systems. Chapter 5 gives the particulars on the verification method and discusses results of its application to the systems described. Chapter 6 summarizes the results of the work.

Chapter 2

AR System Description

Two different optical see-through AR systems were used for the experiments reported in this thesis. The first system is based on a magnetic tracker. The second system uses an infrared video-based tracker. Both systems have stereo capable HMDs, but all experiments were performed in monocular mode. Details for the magnetically tracked system are given in Section 2.1. An overview of the system which uses video-based tracking is given in Section 2.2.

2.1 Magnetically Tracked System

The first system consists of a SGI O2 workstation, a HMD and a magnetic tracker. A photograph of the system is shown in Figure 2.1. A fully detailed description of the calibration requirements and procedures for this system are given in the chapters that follow. Figure 2.2 a schematic for the setup.

The HMD for this system is a Sony Glastron LD-100B. It has a resolution of

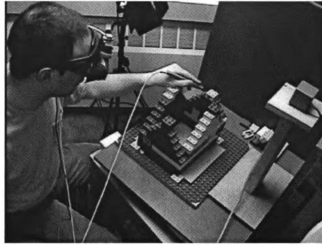


Figure 2.1: A photograph of the magnetically tracked system.

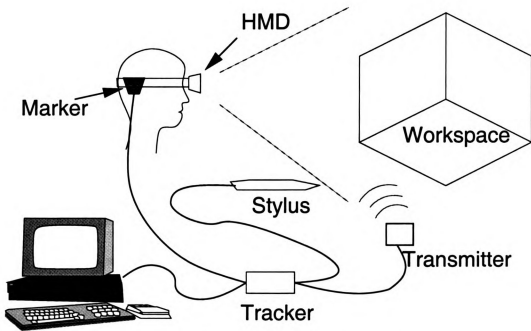


Figure 2.2: Magnetically tracked AR system.

800 × 600 pixels per eye with a refresh rate of 60Hz. The stereo mode is frame-sequential for this display.

The magnetic tracker is a Polhemus Isotrak II with a single transmitter and two receivers: a marker and a digitizing stylus. It is capable of sensing the position and orientation of the receivers at ranges of up to one meter. The marker is rigidly attached to the HMD to track the user's head motion during runtime. The stylus can be used to measure the coordinates of real objects in the workspace at runtime. It can also be used to interact with virtual objects in the workspace.

2.2 Video Tracked System

The second system consists of a SGI 540 Visual Workstation, a HMD, a camera and a graphics tablet. A schematic for this system is shown in Figure 2.3. The details of the tracking system are unknown to this author, so they are not given in this paper [12, 33, 35].

The HMD for this system is a Virtual IO *i-glasses*TM field-sequential stereo display. It has stereo a resolution of 640 × 240 pixels per eye with a refresh rate of 60Hz. The monocular resolution for this display is 640 × 480.

The video-based tracker uses fiducials with known geometry made of retro-reflective material. It uses a camera assembly which is mounted to the HMD. This assembly consists of a CCD camera with an infrared filter and a ring of infrared LEDs mounted around the barrel of the camera. This tracker uses a technique similar to that described in [23] to estimate the pose of the HMD with respect to the fiducials.

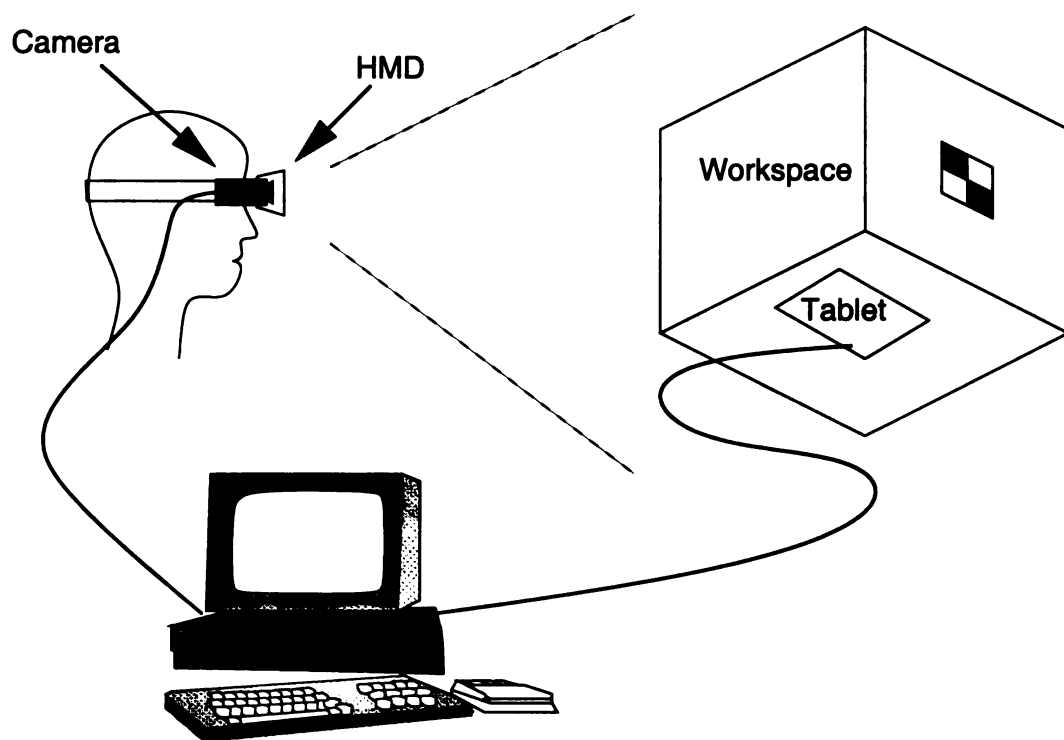


Figure 2.3: AR system with infrared video-based tracking.

The graphics tablet is a Wacom Intuos. It has a $15.25\text{cm} \times 20.25\text{cm}$ writing area.

The resolution for the tablet is 1000 dots per centimeter. There are fiducials mounted on the tablet so that it can be tracked by the camera at runtime.

Chapter 3

Workspace Alignment

While working in a bench-top AR system it is important to have a reference coordinate system which the locations of the real and virtual objects can be specified in. In practice this coordinate system is set in a location which stays fixed during runtime such as a table. While the system is running the user's head must be tracked in some way, *e.g.*, magnetic tracker, video tracker, so that the virtual objects stay aligned with the real objects as the user moves around. Since this motion is measured in the tracker coordinate system (TCS) and the objects are in the workspace coordinate system (WCS), it is necessary to find a transformation between them so that the graphics can be rendered properly.

To align the workspace to the magnetic tracking system, a series of measurements of a calibration object with known position and dimensions is taken. The measurements can be used to find the transformations between the tracker coordinate system (TCS) and the workspace coordinate system (WCS).

The relationship between points in the WCS and the TCS is given by

$$\mathbf{p}_W = s\mathbf{R}\mathbf{p}_T + \mathbf{T}, \quad (3.1)$$

where $\mathbf{p}_W$ and $\mathbf{p}_T$ are the workspace and object points, respectively, s is a scale factor, $\mathbf{R}$ is a 3×3 rotation matrix and $\mathbf{T}$ is the translation vector between the two coordinate system origins. Given a set of points, $\mathbf{p}_{W,i}$, whose WCS locations are known, along with their measured locations in the TCS, $\mathbf{p}_{T,i}$, it is possible to find the transformation, $(s, \mathbf{R}$, and $\mathbf{T})$, with a least squares regression, *i.e.*

$$\text{minimize } \sum_{i=1}^n (\mathbf{p}_{W,i} - s\mathbf{R}\mathbf{p}_{T,i} - \mathbf{T})^2. \quad (3.2)$$

The remainder of this section is dedicated to the detailed analysis of workspace alignment using a magnetic tracker. In sections 3.1 and 3.2, the implementation details for two calibration methods which are based on this model are discussed. Next, a method for removing outliers from the measured data is described in 3.3. Finally, the results of an experimental comparison between the two methods is shown.

3.1 Rigid Alignment

The first alignment method treats the two point clouds as bundles of rays emanating from their centroids. This method makes the assumption that the transform between the two clouds is completely rigid. In order to enforce this rigidity, an orthonormality

constraint must be imposed on $\mathbf{R}$, that is,

$$\mathbf{R}^T \mathbf{R} = \mathbf{I}. \quad (3.3)$$

First, the centroids of both sets of points must be found. They are computed using

$$\bar{\mathbf{p}}_k = \frac{1}{n} \sum_{i=1}^n \mathbf{p}_{k,i}, \quad k \in \{T, W\}. \quad (3.4)$$

Next, both sets of points are converted into bundles of rays about their centroids. This transformation is written as

$$\mathbf{r}_{k,i} = \mathbf{p}_{k,i} - \bar{\mathbf{p}}_k, \quad k \in \{T, W\}. \quad (3.5)$$

From the ray bundles, the scale difference between the two coordinate systems is computed by finding the ratio of the sums of the squared lengths of the individual rays. (This number should be close to 1 if the measured units of the clouds are the same.) It is written compactly as

$$s^2 = \frac{\sum_{i=1}^n \mathbf{r}_{W,i}^2}{\sum_{i=1}^n \mathbf{r}_{T,i}^2}. \quad (3.6)$$

Next, the rotation matrix can be recovered. As previously stated in Equation 3.3, this matrix is constrained to be orthonormal. To begin, the sum of the coordinate-wise products of the corresponding point sets is calculated. This 3×3 matrix, $\mathbf{M}$, is

computed using

$$\mathbf{M} = \sum_{i=1}^n (\mathbf{r}_{W,i})(\mathbf{r}_{T,i})^T. \quad (3.7)$$

Let the matrix $\mathbf{N}$ be defined as

$$\mathbf{N} = \mathbf{M}^T \mathbf{M}. \quad (3.8)$$

Using the definition for $\mathbf{N}$, the rotation matrix, $\mathbf{R}$, can be expressed as

$$\mathbf{R} = \mathbf{M} \mathbf{N}^{-1/2}. \quad (3.9)$$

To find $\mathbf{N}^{-1/2}$, note that $\mathbf{N}$ can be expressed in terms of its eigenvalues, λ_i , and eigenvectors, $\mathbf{v}_i$, as

$$\mathbf{N} = \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T + \lambda_2 \mathbf{v}_2 \mathbf{v}_2^T + \lambda_3 \mathbf{v}_3 \mathbf{v}_3^T. \quad (3.10)$$

After using a standard method to find λ_i and $\mathbf{v}_i$, $\mathbf{N}^{-1/2}$ can be expressed by writing

$$\mathbf{N}^{-1/2} = (\mathbf{M}^T \mathbf{M})^{-1/2} = \frac{1}{\sqrt{\lambda_1}} \mathbf{v}_1 \mathbf{v}_1^T + \frac{1}{\sqrt{\lambda_2}} \mathbf{v}_2 \mathbf{v}_2^T + \frac{1}{\sqrt{\lambda_3}} \mathbf{v}_3 \mathbf{v}_3^T. \quad (3.11)$$

Finally, $\mathbf{T}$ can be found by using $\mathbf{R}$, s and the centroids of the point sets. To compute $\mathbf{T}$, use

$$\mathbf{T} = \bar{\mathbf{p}}_W - s \mathbf{R} \bar{\mathbf{p}}_T. \quad (3.12)$$

Further details of this method are given in [16].

3.2 Affine Alignment

For the affine alignment the orthonormality constraint on $\mathbf{R}$ is relaxed. Formally, it is written as

$$\mathbf{R} = s(\mathbf{R}_o + \mathbf{R}_a), \quad (3.13)$$

where $\mathbf{R}_o$ and $\mathbf{R}_a$ are the orthonormal and affine components of the matrix $\mathbf{R}$. This form builds in a linear approximation to any systematic errors in the tracker measurements. To find the transformation parameters, $\mathbf{R}$ and $\mathbf{T}$, a least squares method is used. After collecting the point cloud data for $\mathbf{p}_W$ and $\mathbf{p}_T$, the block form of the solution can be written as

$$\begin{bmatrix} \mathbf{p}_{M,1}^T & 0 & 0 \\ 0 & \mathbf{p}_{M,1}^T & 0 & \mathbf{I}_{3 \times 3} \\ 0 & 0 & \mathbf{p}_{M,1}^T \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{p}_{M,n}^T & 0 & 0 \\ 0 & \mathbf{p}_{M,n}^T & 0 & \mathbf{I}_{3 \times 3} \\ 0 & 0 & \mathbf{p}_{M,n}^T \end{bmatrix} \begin{bmatrix} \mathbf{R}_1^T \\ \mathbf{R}_2^T \\ \mathbf{R}_3^T \\ \mathbf{T} \end{bmatrix} = \begin{bmatrix} \mathbf{p}_{W,1} \\ \vdots \\ \mathbf{p}_{W,n} \end{bmatrix}, \quad (3.14)$$

where $\mathbf{R}_1$, $\mathbf{R}_2$ and $\mathbf{R}_3$ are the rows of $\mathbf{R}$.

3.3 Robust Alignment Computation

Since both calibration methods are based on least squares, they use all of the point-cloud data in their solutions. Due to noise and inconsistencies in the environment, some of these points may have large errors. These errors reduce the accuracy of the transformation computation considerably if we use traditional least-squares methods, especially in the rigid alignment procedure. These errors, in turn, propagate into the HMD calibration procedure.

To detect and remove points which have large errors, the Random Sample Consensus (RANSAC) method described in [9] is used. The RANSAC algorithm is shown in Algorithm 1. The inputs to the algorithm are the data, the number of subsets to try, k , the minimum subset size, t , and the error threshold, ϵ . This section describes the choices made for each in detail.

1. Select a subset of size t or larger from the input data.
2. Solve the system and evaluate the error $\epsilon = \|\mathbf{Ax} - \mathbf{b}\|$.
3. If the error ϵ is less than some prescribed tolerance, use $\mathbf{x}$ as the solution and terminate.
4. If the number of subsets evaluated is larger than some threshold, terminate and report failure, otherwise goto step 1.

Algorithm 1: RANSAC Algorithm

To begin, an error tolerance must be determined for points coming from the tracker. There are two main characteristics which need to be quantified. The first is the overall tracker measurement variance. This quantity describes the amount of variance expected when measuring the same point multiple times. It is determined

Table 3.1: Pointwise standard deviation measurements.

Point	Control σ (cm)	Warped σ (cm)
1	0.04	0.03
2	0.04	0.05
3	0.06	0.03
4	0.12	0.03
5	0.06	0.04
6	0.05	0.05
7	0.03	0.08
8	0.07	0.04
9	0.06	0.06
10	0.03	0.15
11	0.05	0.05
12	0.05	0.05
Mean:	0.05	0.06

by repeatedly measuring the calibration jig calculating the standard deviation at each point. The results for these measurements are tabulated in Table 3.1. The left hand column shows the standard deviations for measurements at 12 locations on the calibration fixture in a magnetically “clean” environment. The right hand column shows the standard deviations for measurements of the same points taken with a large metal object in the workspace to distort the magnetic field. Notice that the two columns are essentially the same. This indicates that static metallic objects induce a fixed amount of warpage to the field. Most of the measurements fall within the manufacturer specified standard error of 1mm.

While the pointwise standard deviation is interesting, it does not capture the warpage of the field.

In order to see the distortions in the workspace it is necessary to examine the relationships *between* the points. This is accomplished by computing the distances

Table 3.2: Magnetic field distortion information.

Data Set	μ_{pos} (cm)	σ_{pos} (cm)
Control	0.32	0.15
Noisy	1.02	2.33

between all of the pairs of measured points (on the calibration fixture) and comparing them to their known values. This will give estimates of the errors in length which in turn can be used to show distortions in the magnetic field.

The length errors from the previously described sets of data are shown in Figure 3.1. The average errors along with their standard deviations provide an estimate of the expected measurement error for the points used during a calibration. These values are shown in Table 3.2. The distribution of the errors is shown in Figure 3.2. Assuming a normal error distribution, the value for ϵ is given by the mean error plus 1 standard deviation, *i.e.*, $\epsilon = 0.47\text{cm}$.

These results confirm that distortions from metallic objects are constant and could be calibrated out of a system [19]. To perform such a calibration, it would be necessary to create a polynomial map for the antenna, *i.e.*, calibrate the field. This map could be used to warp the measured positions of points during calibration and runtime.

The second free parameter for the RANSAC procedure is k , the maximum number of attempts at finding a consensus set. Let w represent the probability that a measured point is within the error tolerance ϵ . The expected number of trials, k , to select n acceptable data points can be written

$$E(k) = w^n + 2(1 - w^n)w^n + 3(1 - w^n)^2w^n + \cdots + i(1 - w^n)^{i-1}w^n + \cdots, \quad (3.15)$$

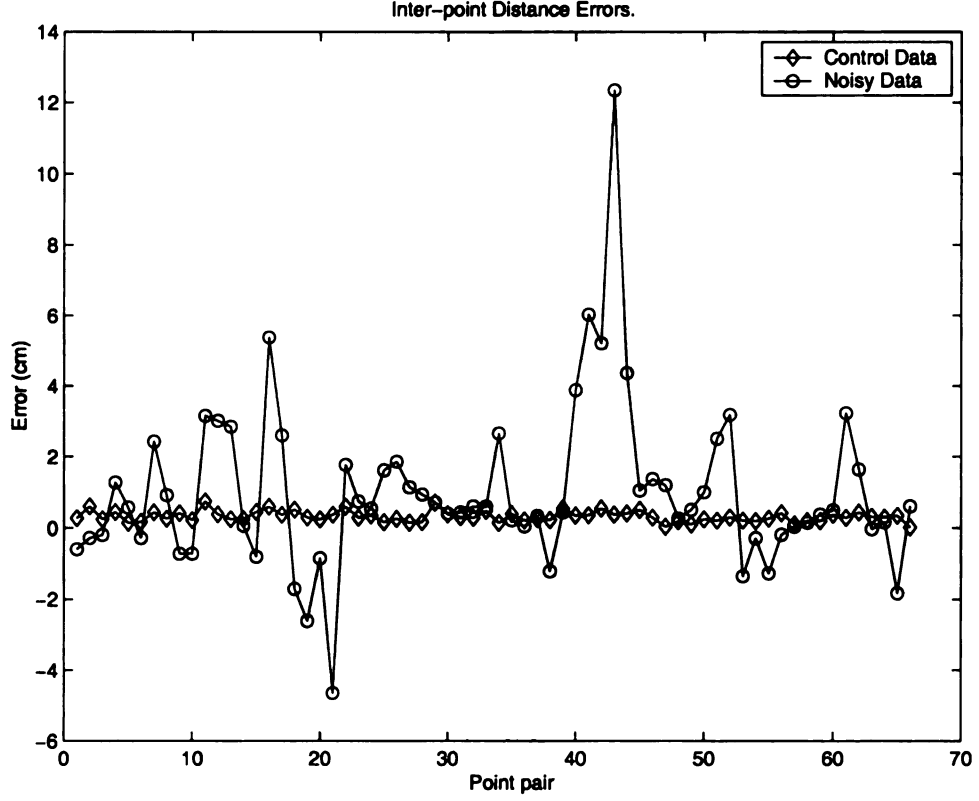


Figure 3.1: Pairwise inter-point differences for measured points.

which can be simplified to

$$E(k) = w^n(1 + 2(1 - w^n) + 3(1 - w^n)^2 + \dots + i(1 - w^n)^{i-1} + \dots). \quad (3.16)$$

It is possible to simplify the series in Equation 3.16 by using the identity for the sum of a geometric series

$$\frac{r}{1 - r} = r + r^2 + r^3 + \dots + r^i + \dots. \quad (3.17)$$

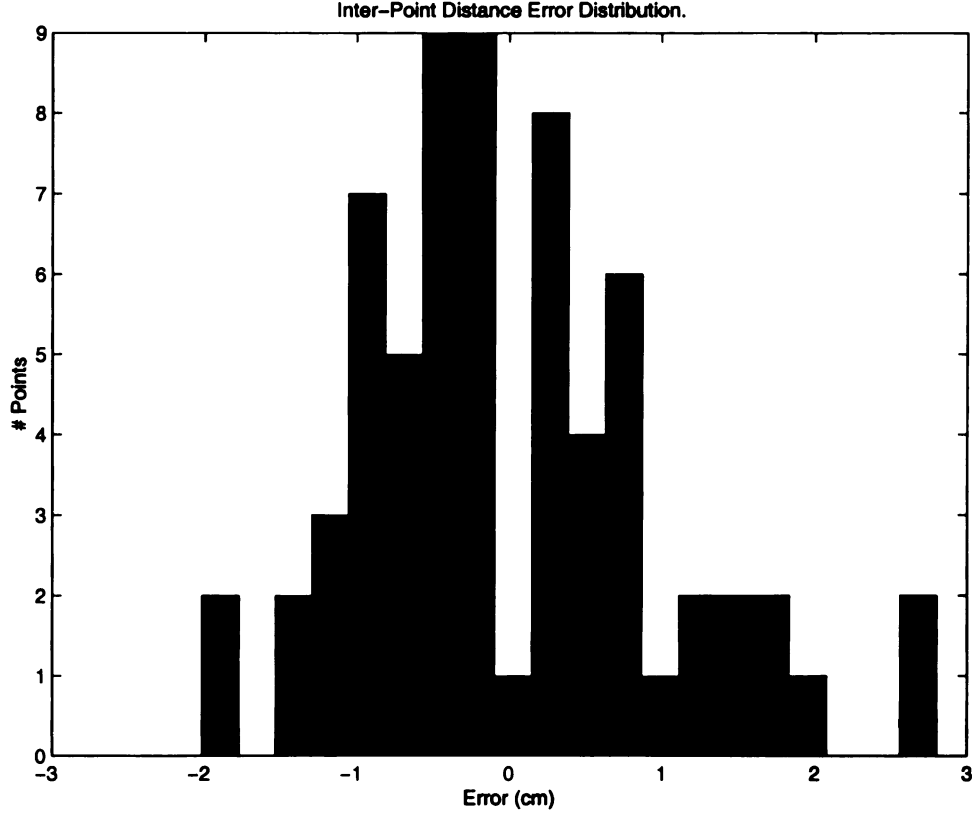


Figure 3.2: Distribution of inter-point distances is approximately normal.

Differentiating this series with respect to r produces

$$\frac{1}{(1-r)^2} = 1 + 2r + 3r^2 + \dots + ir^{i-1} + \dots \quad (3.18)$$

Substituting $1 - w^n$ for r in the above gives

$$\frac{1}{(w^n)^2} = 1 + 2(1 - w^n) + 3(1 - w^n)^2 + \dots + i(1 - w^n)^{i-1} + \dots, \quad (3.19)$$

therefore,

$$E(k) = \frac{1}{w^n}. \quad (3.20)$$

Assuming that the measurement errors are normally distributed, a point will be within

the error tolerance 68% of the time. Thus, if at least $n = 4$ points are needed to find an alignment, then $E(k) = 1/(0.68)^4 = 5$. Three times this value will remove outliers about 99.6% of the time.

The final parameter which needs to be specified is the minimum size of the consensus set, t . If p is the probability that a data point is within the error tolerance of an incorrect model, then p^{t-n} is the probability that a consensus set has converged on an incorrect solution for the model. Thus, the value of p^{t-n} should be small. Even though there is no good way to establish p *a priori*, it is reasonable to assume that $p < w$. Setting $p = 0.5$ and requiring a minimum confidence level of 95% forces $t - n$ to be at least 5. This means that $t = 9$ and the probability of selecting an incorrect solution is 3.12%.

3.4 Experimental Results

To test the two algorithms, two sets of data, consisting of 120 points per set, were generated by measuring 12 points on the calibration object ten times each. The first set, which shall be referred to as the control set, was taken in a magnetically clean environment. The second set, which will be called the noisy set, was taken with a large aluminum block placed near the center of the workspace in order to distort the magnetic field. The test data sets were used to perform two groups of experiments comparing the least-squares and robust versions rigid and affine methods described in the previous sections of this chapter. The results are shown in Table 3.3.

For each of the experiments, two subsets of size 12 were randomly selected and

used for calibration. The resulting matrix was then used to transform the entire data set. The transformed points were compared to the actual locations and error statistics were computed. This procedure was repeated five times for each of the combinations of the clean and noisy data sets, *e.g.*, calibrate with clean data, transform clean data, calibrate with clean data, transform noisy data, etc. Table 3.3 summarizes the results of the 16 experiments. In this table, the μ column represents the average error produced by the standard least-squares calibration method, the Robust μ represents the average error from the robust procedure and the σ and Robust σ columns show the standard deviations of the two procedures, respectively.

There are four important points to note from Table 3.3. The first is that the affine methods tend to have better error statistics than the rigid methods. This comes from the lack of constraints on the affine calibration matrix which allows for a linear approximation to the distortion in the magnetic field. The second noteworthy point is that the least-squares and robust methods are nearly identical for the experiments which were calibrated using clean data. This is to be expected since there were no outliers in the data. On the other hand, the robust methods worked significantly better when noisy data was used for calibration. Again, this is to be expected since the outliers will be removed by the robust procedures. Finally, the robust methods showed higher variance on average. This is because the outliers were not removed from the re-projected data.

Figures 3.3–3.6 show comparisons between the least-squares and robust methods for all combinations of the data sets. Figure 3.3 shows the errors for the affine calibration procedures. In this case, the errors for the least-squares and robust procedures

Table 3.3: Error analysis results.

Experiment	Statistics			
	μ (cm)	Robust μ (cm)	σ (cm)	Robust σ (cm)
Affine Control Calibration Control Data	0.13	0.13	0.05	0.06
Affine Control Calibration Noisy Data	2.58	2.58	2.60	2.60
Affine Noisy Calibration Control Data	2.72	2.03	1.26	1.21
Affine Noisy Calibration Noisy Data	1.94	1.90	1.30	2.27
Rigid Control Calibration Control Data	0.15	0.15	0.07	0.07
Rigid Control Calibration Noisy Data	2.55	2.55	2.60	2.60
Rigid Noisy Calibration Control Data	1.54	1.20	0.66	0.55
Rigid Noisy Calibration Noisy Data	2.34	2.19	2.14	2.51

are nearly identical.

Figure 3.4 shows the results for the affine calibration with noisy data. Note that the robust procedure has generally smaller errors, but greatly magnifies the errors from the outliers.

Figure 3.5 shows the results of the rigid calibration method using clean data. Again, there is not a significant difference between the least-squares and robust procedures.

Figure 3.6 shows the results from the rigid calibration performed with noisy data. The rigid calibration scheme does quite well when transforming the clean data set, but magnifies the outliers considerably when transforming the noisy set.

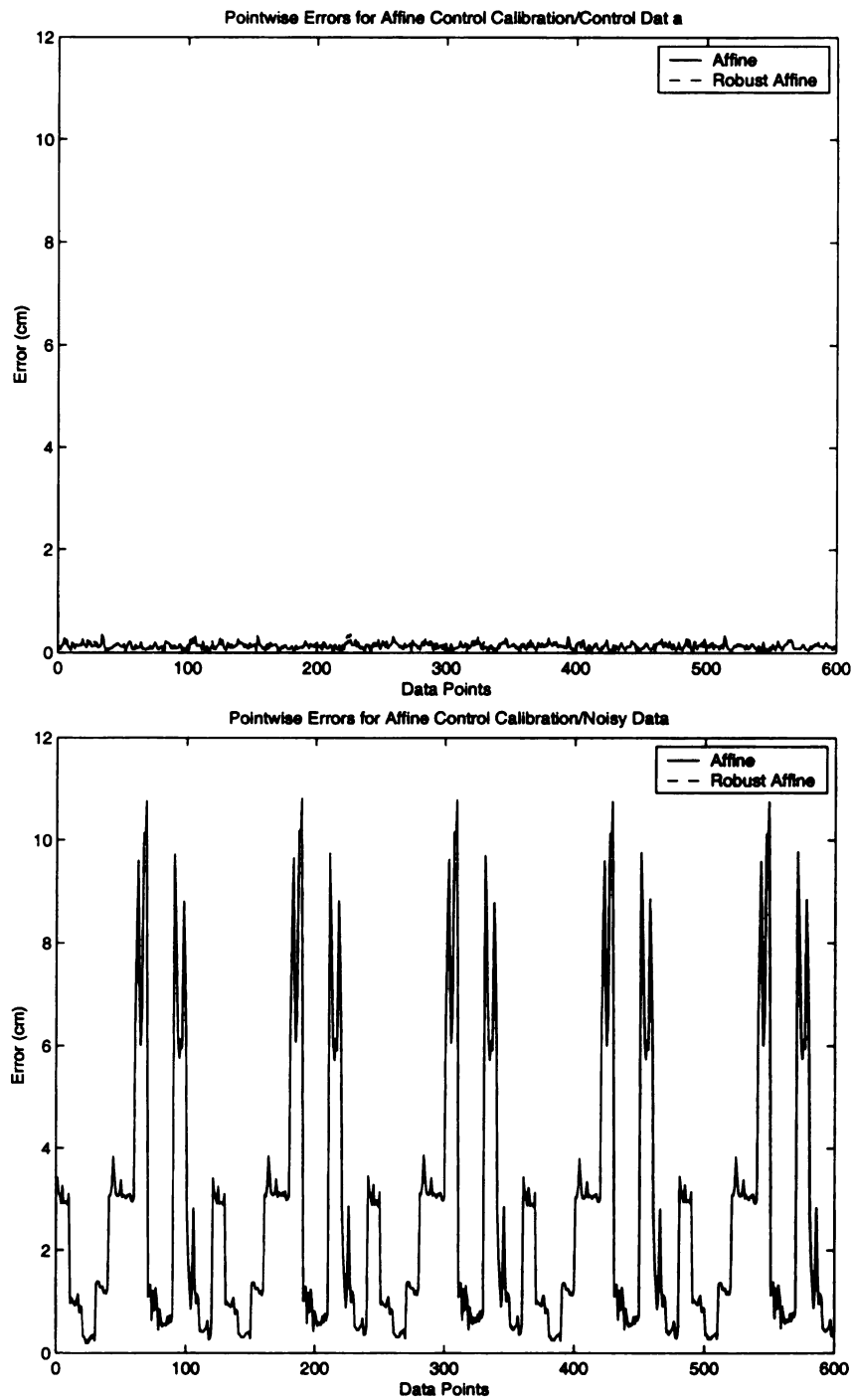


Figure 3.3: Errors using the affine methods with control calibration data.

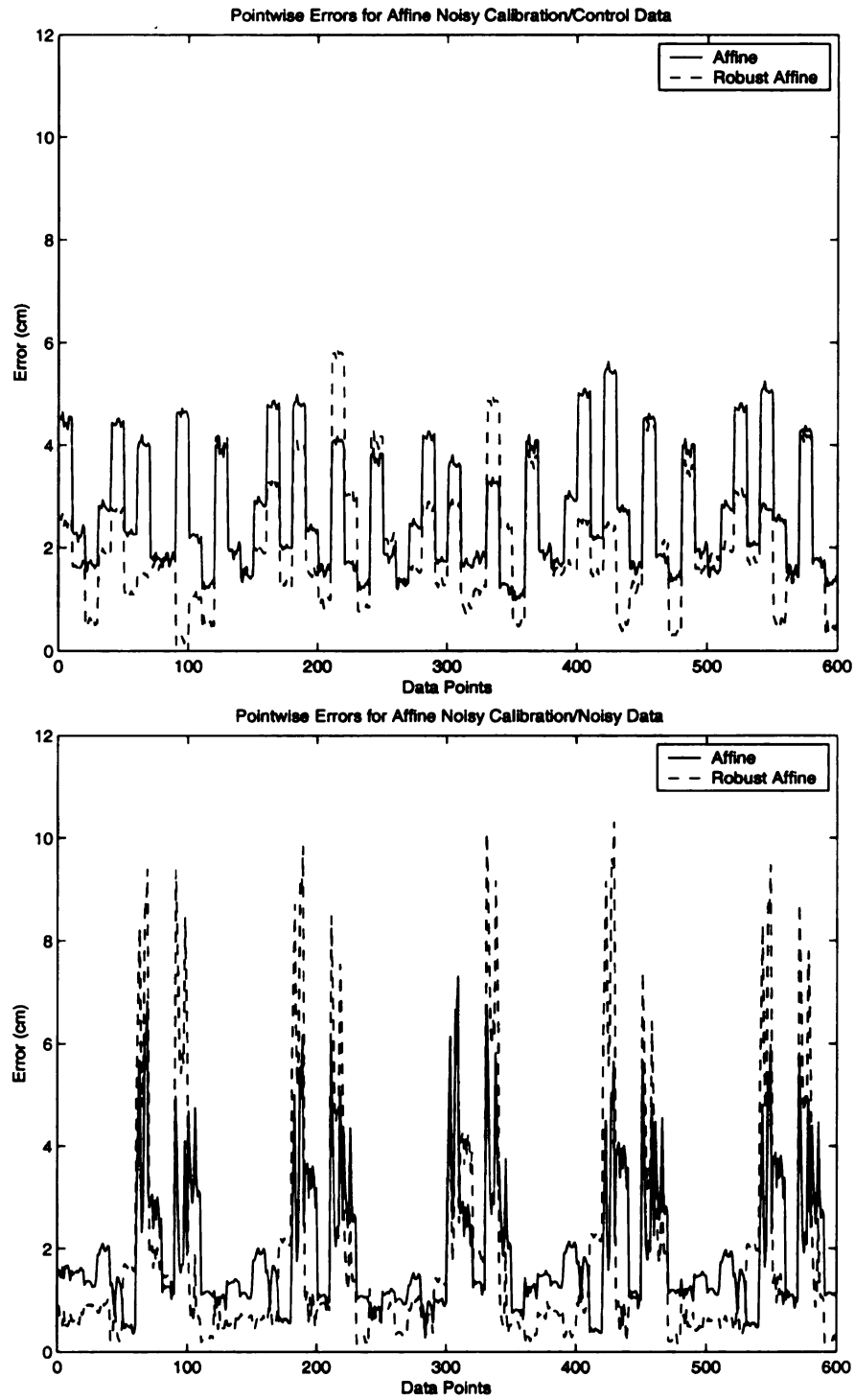


Figure 3.4: Errors using the affine methods with noisy calibration data. The outliers are highly visible in the robust procedure.

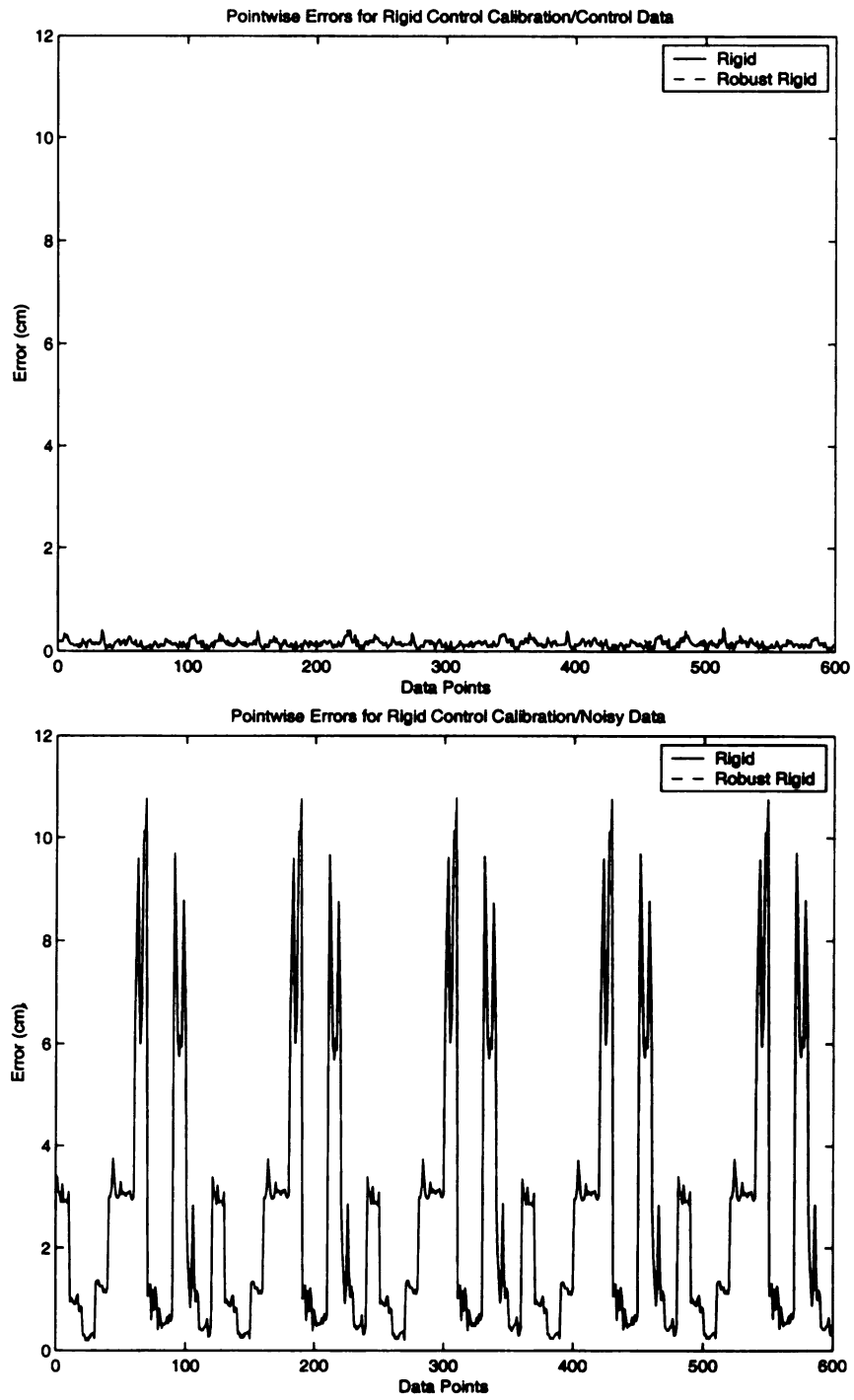


Figure 3.5: Errors using the rigid methods with control calibration data.

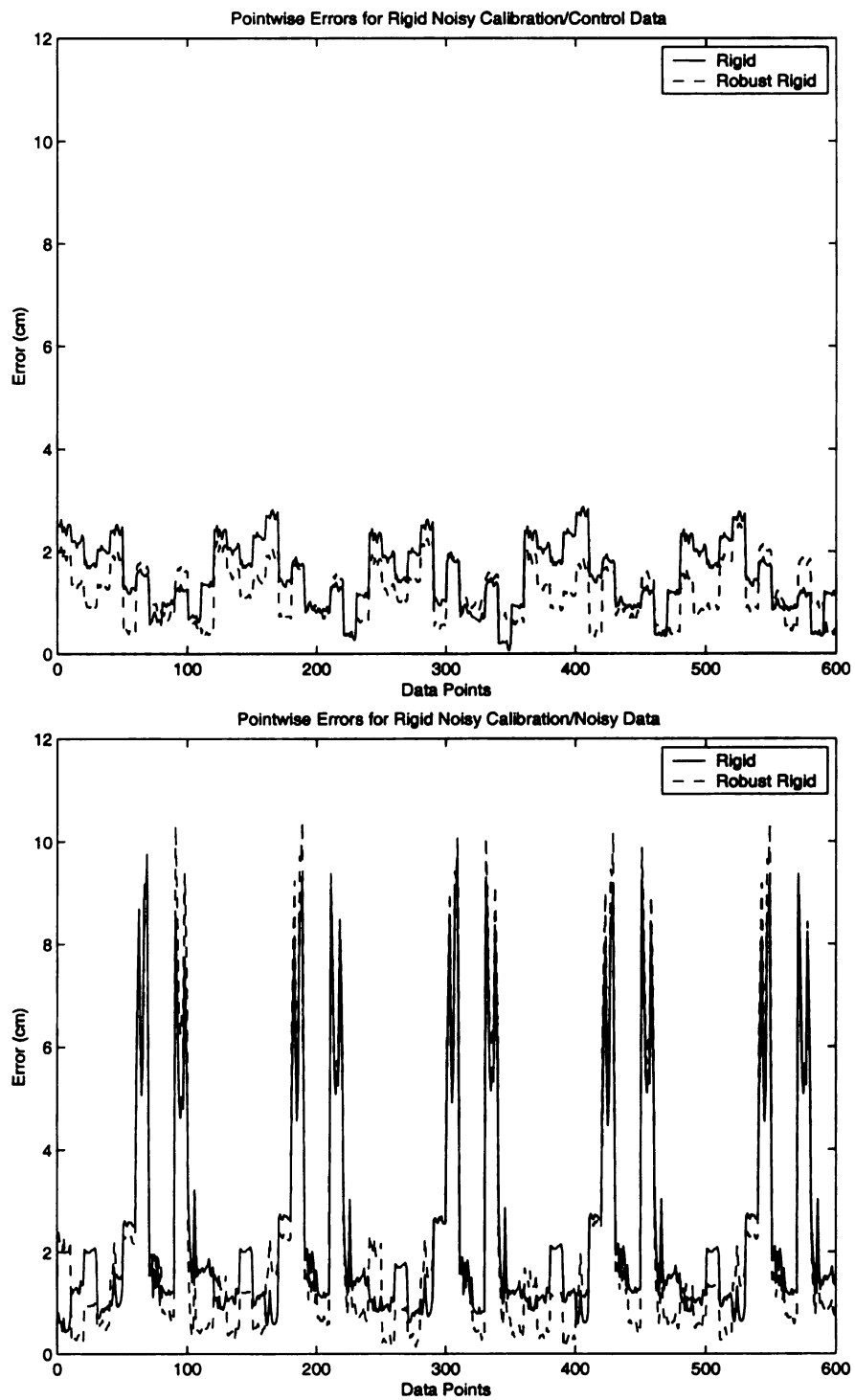


Figure 3.6: Errors using the rigid methods with noisy calibration data.

Chapter 4

HMD Calibration

This section gives the particulars for the calibration of an optical see-through head-mounted display (HMD). The primary assumption behind the method is that the eye/HMD system is a pinhole camera. There are several well known algorithms for calibrating pinhole cameras, but most of them will not work in the context of HMDs because they require the camera to be rigidly mounted. The method given herein has no such requirement.

The rest of this chapter is divided into three sections. The first section describes the camera model. The second section gives a numerical solution to the calibration problem. The final section tells of the implementation details for the system which is currently in place.

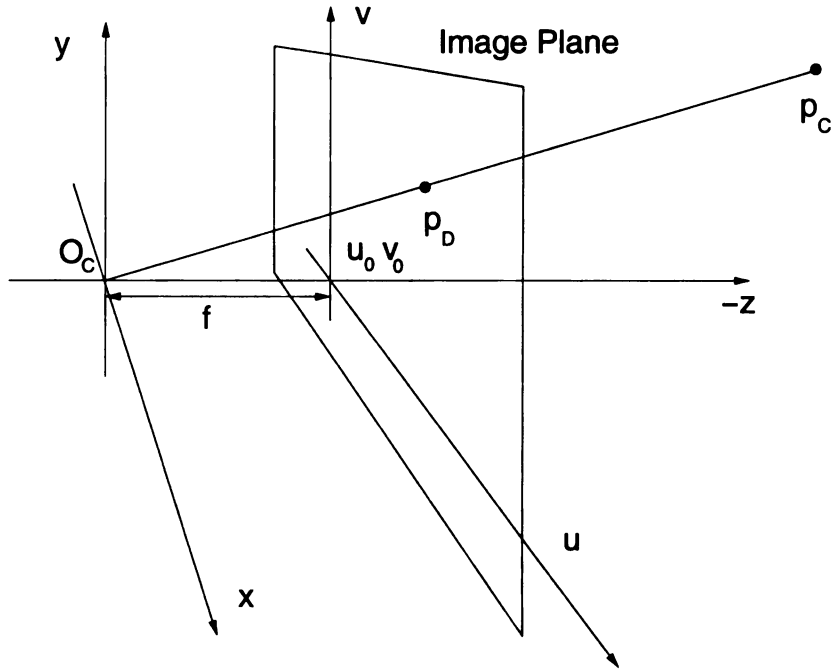


Figure 4.1: The geometry of a simple pinhole camera model for perspective projection.

4.1 Camera Model

A simple pinhole camera (Figure 4.1) is used as a model for the HMD [7]. This model defines the basic projective imaging geometry with which 3D objects in the system are projected onto the 2D display surface. This is an idealized model commonly used in computer graphics and computer vision to capture imaging geometry. It does not account for non-linear optical effects (such as radial distortions) that are often properties of real cameras. The model uses a right-handed coordinate system with the center of projection at the origin and the image plane at a distance of f (focal length) from the origin. Views through the center of projection are assumed to be views down the $-z$ axis.

The image of a visible point in the camera coordinate system is formed by passing

a ray through the point, $\mathbf{p}_C$, and the center of projection, O_C , and computing the intersection of this ray with the image plane, $\mathbf{p}_D$. If $\mathbf{p}_C = [x_c, y_c, z_c, 1]^T$ and $\mathbf{p}_D = [u, v, 1]^T$ then using similar triangles the display coordinates can be written,

$$\begin{aligned} u &= f s_u x_c / z_c + u_0, \\ v &= f s_v y_c / z_c + v_0, \end{aligned} \tag{4.1}$$

where (u_0, v_0) are the coordinates of the image plane origin in the camera coordinate system, and s_u and s_v account for the pixel sizes of the display. Let $f_u = s_u f$ and $f_v = s_v f$. Placing these values into matrix form produces a transformation $\mathbf{P}$ which is used to transform 3-dimensional homogeneous coordinates, $(\mathbf{p}_C)$, into 2-dimensional homogeneous display coordinates, $(\mathbf{p}_D)$. $\mathbf{P}$ is given by

$$\mathbf{P} = \begin{bmatrix} f_u & 0 & u_0 & 0 \\ 0 & f_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{4.2}$$

In practice, the u and v axes may not be perpendicular to each other. To correct for this, a skew parameter must be added to $\mathbf{P}$ to form a new matrix $\mathbf{Q}$, such that,

$$\mathbf{Q} = \begin{bmatrix} f_u & \tau & u_0 & 0 \\ 0 & f_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{4.3}$$

where τ represents the skew between the u and v axes. Note that since $\mathbf{Q}$ is projective,

it is defined only up to a scale factor w . Putting everything together, a point's display coordinates can be found by

$$\begin{bmatrix} wu \\ wv \\ w \end{bmatrix} = \begin{bmatrix} f_u & \tau & u_0 & 0 \\ 0 & f_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_C \\ y_C \\ z_C \\ 1 \end{bmatrix}, \quad (4.4)$$

where w is the perspective scale parameter related to the depth ($w = z_C$).

In general, it is not possible to locate the HMD coordinate system directly. Instead, there is usually a tracking system rigidly attached to the HMD whose workspace coordinates can be measured in real time. To account for this, an additional transformation between the tracking system and the HMD must be estimated. This transformation is rigid and can be written

$$\mathbf{T} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (4.5)$$

where the r_{ij} are the elements of a 3×3 rotation matrix and the t_i are the translational components.

Multiplying $\mathbf{Q}$ by $\mathbf{T}$ produces the transformation which must be applied to a point to get it from the tracker coordinate system into the HMD coordinate system and

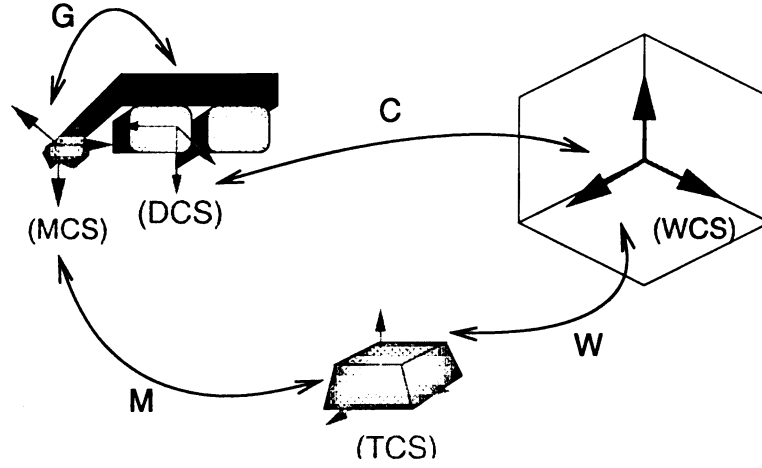


Figure 4.2: Coordinate transformations used for display calibration.

onto the image plane. This is the transformation **G** in Figure 4.2. It is

$$\mathbf{G} = \mathbf{QT} = \begin{bmatrix} g_{11} & g_{12} & g_{13} & g_{14} \\ g_{21} & g_{22} & g_{23} & g_{24} \\ g_{31} & g_{32} & g_{33} & g_{34} \end{bmatrix}. \quad (4.6)$$

The individual components of **G** are

$$\begin{aligned} g_{11} &= f_u r_{11} + \tau r_{21} + u_0 r_{31}, & g_{21} &= f_v r_{21} + v_0 r_{31}, & g_{31} &= r_{31}, \\ g_{12} &= f_u r_{12} + \tau r_{22} + u_0 r_{32}, & g_{22} &= f_v r_{22} + v_0 r_{32}, & g_{32} &= r_{32}, \\ g_{13} &= f_u r_{13} + \tau r_{23} + u_0 r_{33}, & g_{23} &= f_v r_{23} + v_0 r_{33}, & g_{33} &= r_{33}, \\ g_{14} &= f_u t_1 + \tau t_2 + u_0 t_3, & g_{24} &= f_v t_2 + v_0 t_3, & g_{34} &= t_3. \end{aligned} \quad (4.7)$$

Figure 4.2 shows the transformations a point in the workspace coordinate system

must undergo to get into the display coordinate system. First, points are transformed from the WCS into the TCS using the transformation $\mathbf{W}$ found during the tracker calibration step described in Section 3. Then they go from the TCS into the MCS via the $\mathbf{M}$ transformation which is updated by the tracker at each timestep of the system. The entire transformation from the WCS to the MCS is given by

$$\mathbf{p}_M = \mathbf{M}\mathbf{W}\mathbf{p}_W, \quad (4.8)$$

where $\mathbf{p}_M$ is a point in the tracker mark coordinate system and $\mathbf{p}_W$ is a point in the workspace coordinate system.

Next, the point $\mathbf{p}_M$ needs to pass through the transformation from the tracker coordinate system into the HMD coordinate system. This transformation is written

$$\mathbf{p}_D = \mathbf{G}\mathbf{p}_M. \quad (4.9)$$

It is this transformation, $\mathbf{G}$, which must be estimated during calibration. Expanding Equation 4.9 and dividing by the homogeneous scale factor w produces the pixel coordinates for any visible point:

$$\begin{aligned} u_d &= \frac{u}{w} = \frac{g_{11}x_m + g_{12}y_m + g_{13}z_m + g_{14}}{g_{31}x_m + g_{32}y_m + g_{33}z_m + g_{34}}, \\ v_d &= \frac{v}{w} = \frac{g_{21}x_m + g_{22}y_m + g_{23}z_m + g_{24}}{g_{31}x_m + g_{32}y_m + g_{33}z_m + g_{34}}, \end{aligned} \quad (4.10)$$

where u_d and v_d are the pixel coordinates of the point.

4.2 Estimation of the Display Parameters

By rearranging Equation 4.10 and it is possible to write two constraints on the $\mathbf{G}$ transformation,

$$\begin{aligned} (g_{11}x_m + g_{12}y_m + g_{13}z_m + g_{14}) - u_d(g_{31}x_m + g_{32}y_m + g_{33}z_m + g_{34}) &= 0, \\ (g_{21}x_m + g_{22}y_m + g_{23}z_m + g_{24}) - v_d(g_{31}x_m + g_{32}y_m + g_{33}z_m + g_{34}) &= 0. \end{aligned} \quad (4.11)$$

Since there are 12 parameters to solve for, at least 6 sets of 2D/3D point correspondences must be collected by the user. If the i^{th} tuple of correspondences is defined as $(x_{m,i}, y_{m,i}, z_{m,i}, u_{d,i}, v_{d,i})$ then the solution for $\mathbf{G}$ can be found by estimating a solution to the homogeneous equation

$$\mathbf{B}\mathbf{g} = 0, \quad (4.12)$$

where $\mathbf{g}$ is a vector made by concatenating the transposed rows of $\mathbf{G}$ *i.e.*

$$\mathbf{g} = [g_{11}, g_{12}, g_{13}, g_{14}, g_{21}, g_{22}, g_{23}, g_{24}, g_{31}, g_{32}, g_{33}, g_{34}]^T, \quad (4.13)$$

and

$$\mathbf{B} = \begin{bmatrix} \mathbf{p}_{M,1}^T & 0 & -u_{d,1}\mathbf{p}_{M,1}^T \\ 0 & \mathbf{p}_{M,1}^T & -v_{d,1}\mathbf{p}_{M,1}^T \\ \vdots & \vdots & \vdots \\ \mathbf{p}_{M,n}^T & 0 & -u_{d,n}\mathbf{p}_{M,n}^T \\ 0 & \mathbf{p}_{M,n}^T & -v_{d,n}\mathbf{p}_{M,n}^T \end{bmatrix}. \quad (4.14)$$

Solving Equation 4.12 will produce the $\mathbf{G}$ matrix. Because this equation is homogeneous, there must be another constraint to avoid a null solution. This constraint can be made by remembering that $\mathbf{G}$ is only defined up to a scale factor. This means that it is possible to solve for the $\mathbf{g}$ vector using the eigensystem of $\mathbf{B}^T\mathbf{B}$. Formally, it is written

$$\text{minimize } \|\mathbf{B}\mathbf{g}\|^2 \text{ subject to } \|\mathbf{g}\| = 1. \quad (4.15)$$

The $\mathbf{g}$ vector corresponds to the eigenvector of $\mathbf{B}^T\mathbf{B}$ with the smallest associated eigenvalue [7]. Once this solution has been found it can be placed into $\mathbf{G}$. If the sign of the determinant of $\mathbf{G}(1 : 3, 1 : 3)$ is greater than zero, then it is necessary to negate $\mathbf{G}$ [33]. This is due to a second solution which exists behind the camera.

4.3 Calibration Procedure

Practicality is an important feature of any HMD calibration algorithm. It needs to be simple to use and accurate. A first attempt at a collection algorithm would ask the user to align crosshairs in the image plane with points in the workspace in some predetermined order. While correct, it can be confusing for the user as they must remember the sequence, or someone must direct them. With a small simplification, the user is freed of this burden. Instead of being asked to align different points in the workspace, they are asked to align the same point repeatedly from different positions. This technique is known as Single Point Active Alignment Matrix (SPAAM) calibration [33]. A compact description of the procedure is given in Algorithm 2.

1. A single point is selected in the workspace coordinate system.
2. The user is presented with a crosshair in the display plane.
3. The user moves their head until the crosshair is aligned with the point in the workspace.
4. The user clicks the mouse to indicate they are aligned and the point's location in the marker coordinate system is computed and recorded.
5. The cursor is moved to a different location on the display plane and the process is repeated starting with step 2 until enough points have been collected.
6. The calibration matrix is estimated and the process terminates.

Algorithm 2: SPAAM algorithm used for calibration of the HMD.

Chapter 5

Evaluating a Calibration

When designing a scheme to evaluate an AR system, there are many factors to consider. First, the procedure must be on-line. An AR system is not complete without its human user. Schemes which replace the user with a camera and a rigid mounting fixture do not reflect the situation properly. In fact, much of the error observed in an AR system is from the user. During HMD calibration, for instance, if the user raises their eyebrows, the display moves with respect to the eye and error is introduced into the system.

Another important observation has to do with the system complexity. The amount of error a user introduces into the system is reflected in the difficulty of the calibration and evaluation tasks. Several of the users tested remarked that it was difficult to keep the crosshair on the alignment point during the HMD calibration (see Section 4.3).

A third feature an evaluation scheme must exhibit is precision. It must provide consistent quantitative results from trial to trial. These results must give a clear indication of the accuracy that a user will perceive. It must be able to communicate

this accuracy to the user through some feedback mechanism. With this and the two points above in mind, the evaluation scheme can be laid out.

Consider an augmented reality system with an optical see-through HMD and a real-time tracking system. Assuming that the user has calibrated the system along with the HMD, graphics can be overlaid onto the real workspace with some unknown accuracy. The aim is to measure the registration accuracy between the virtual objects and the real objects. The results from these measurements can be used in two different ways. First, calibration accuracy information can be obtained. This information can be used to determine the expected registration accuracy at a given distance. For instance, the system could report, “At 70 cm away from the target, the user can correctly identify items which are 1 cm apart.” This is useful in determining if the calibration is good enough for a particular application. Second, structured errors in the tracking system can be identified and possibly corrected. Structured errors are errors in the system which have a regular form, *e.g.*, linear, quadratic, etc. For instance, problem areas in a magnetic tracker can be found.

The rest of this chapter describes a method for evaluating the accuracy of an optical see-through AR system at runtime. The algorithm is given in Section 5.1. Section 5.2 discusses the error metric used and its implications. Experimental results are presented and discussed in Section 5.3. An analysis of the structure of the errors is presented in Section 5.4. Conclusions about the system evaluated are drawn in Section 5.5.

5.1 Evaluation Algorithm

The evaluation module is comprised of an optical see-through AR system and an input device capable of reporting positions in the workspace coordinate system. For a video-based tracking system, this device may be a graphics tablet with fiducials on it so it can be tracked. In the case of a magnetic tracking system, the tracker's stylus can be used. Figure 5.1 shows a schematic for a system with a video based tracker.

Any time after system calibration has been performed, the user may invoke the evaluation module. The evaluation module performs the steps shown in Algorithm 3. Note that since the system is running during the process, the user's head is being tracked so they are free to move around during the validation process.

1. Superimpose a virtual object, *e.g.* a small sphere, onto a known position on the board.
2. Ask the user to touch the board where they perceive the object.
3. Move the object to another location and goto step 1 until the area of the board is evenly covered. Once the area has been covered goto step 4.
4. Compute and display the system error.

Algorithm 3: Evaluation module algorithm.

5.2 Planar Calibration Validation

While an AR system is running, it may be important for the user to get an estimate of their current calibration accuracy. For instance, if the HMD is moved with respect to the user's eye, then the calibration could be invalidated. If this situation occurs,

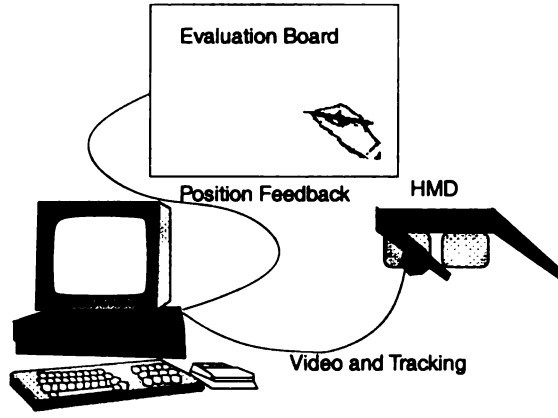


Figure 5.1: The optical see-through evaluation system.

the module presented in Section 5.1 can be invoked.

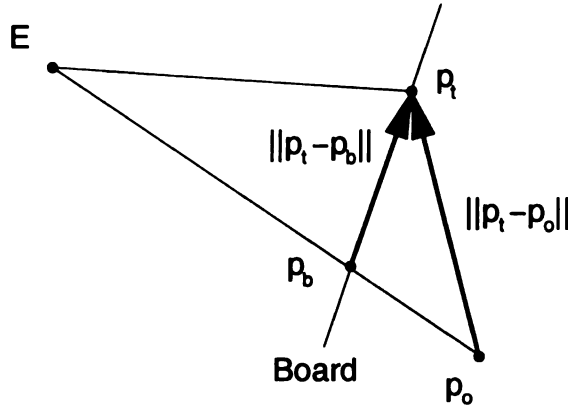


Figure 5.2: Error is measured in the plane of the evaluation board as seen from the eyepoint, E . The true location is $\mathbf{p}_t$, the observed location is $\mathbf{p}_o$ and the projection of the observed point onto the board is $\mathbf{p}_b$.

For validation, there are three points of interest $\mathbf{p}_t$, $\mathbf{p}_o$, and $\mathbf{p}_b$. The first point, $\mathbf{p}_t$, is the theoretical location of the virtual point in the workspace coordinate system. This point should be unknown to the user so there is no bias in the results. The theoretical point is rendered in the display using the calibration parameters of the system. This produces the second point, $\mathbf{p}_o$. $\mathbf{p}_o$ is the location where the user

perceives the point to be in the workspace. In principle, it is impossible for the user to tell the 3-dimensional location of the point $\mathbf{p}_o$. If the HMD and tracker calibrations are exact, then $\mathbf{p}_t$ and $\mathbf{p}_o$ will coincide. As this is not usually the case, a final point, $\mathbf{p}_b$ must be introduced. $\mathbf{p}_b$ is the projection of $\mathbf{p}_o$ into the plane of the evaluation board. This point is selected by the user when they touch the board during system validation. Figure 5.2 shows the geometry of the 3 points.

It is important to note that the planar error metric completely depends on the user's viewpoint. Figure 5.3 illustrates this point. The observed error from the viewpoint E is larger than that of E' . Also notice that from multiple views it is possible to estimate the location of $\mathbf{p}_o$ through the use of point reconstruction techniques. This issue is addressed in Section 5.4

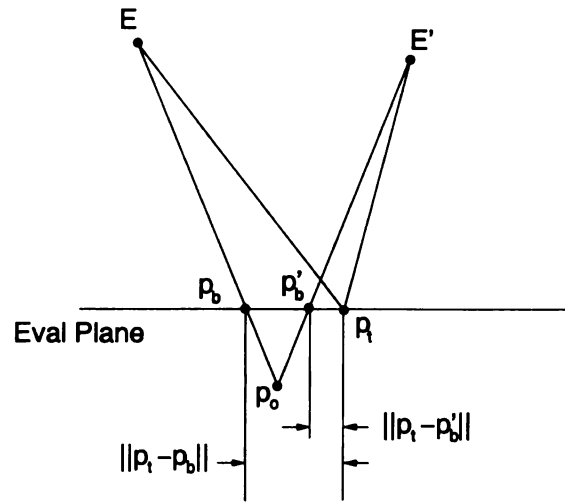


Figure 5.3: Error in the plane of the evaluation board as seen from the eye-points, E and E' . The error $\|\mathbf{p}_t - \mathbf{p}_b\|$ is larger than $\|\mathbf{p}_t - \mathbf{p}'_b\|$.

Once the data has been gathered, the evaluation module computes the Euclidean distances between each of the theoretical points and their projected observed loca-

tions, *i.e.*, $\|\mathbf{p}_t - \mathbf{p}_b\|$, as shown in Figure 5.2. The statistics from these computed distances can be used as estimates of tolerances for the current calibration. The errors could also be used to give more direct feedback, such as drawing spheres at each of the picked locations with colors and/or radii corresponding to the amount of error.

5.3 Experimental Results for Planar Verification

In this section, results from two different AR systems are presented. The first system uses a magnetic tracker; the second uses an infrared video-based tracker. Both systems have monocular displays which were calibrated with the method described in Chapter 4. After calibration, the users invoked the evaluation module described in Section 5.1.

The results from a typical user of the magnetically tracked system are shown in Figure 5.4. This plot shows the theoretical locations as circles along with their observed locations from three different zones in the workspace. These zones are defined as the 3-dimensional regions to which the users' heads were confined when the observations were made. The range of head motion was restricted because the users were seated during the evaluation procedures. Figure 5.5 shows an example of the three dimensional viewing zones for an evaluation run.

The theoretical points were placed in the plane $z = -0.63cm$. The users recorded observations by clicking the observed location with the stylus from the magnetic tracker. The observations of the points from the three viewing zones are indicated on the plot by plusses, stars and triangles. The mean and maximum errors for each zone and the global errors are tabulated in Table 5.1. Tables 5.2 and 5.3 show the mean

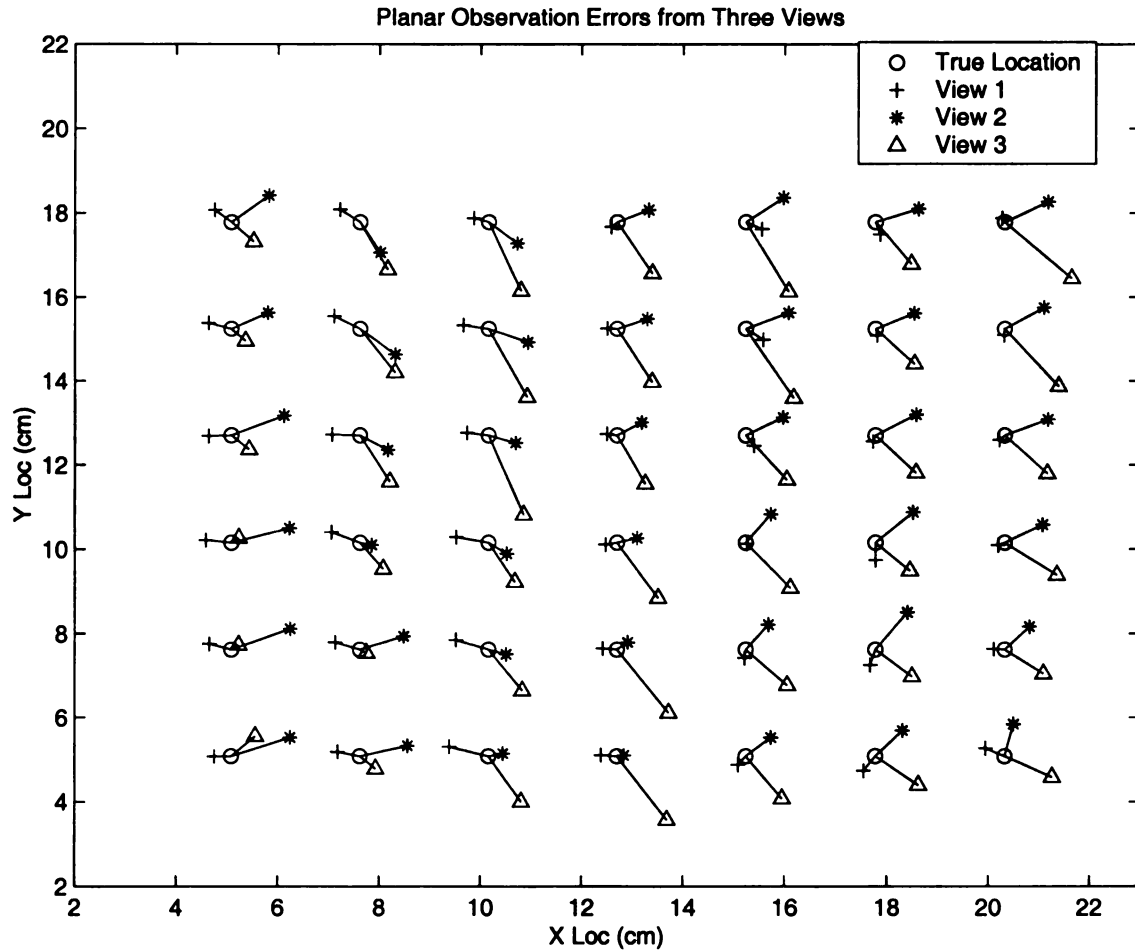


Figure 5.4: Error fields for an AR system which uses a magnetic tracker. These are the planar errors as observed from three zones in the workspace.

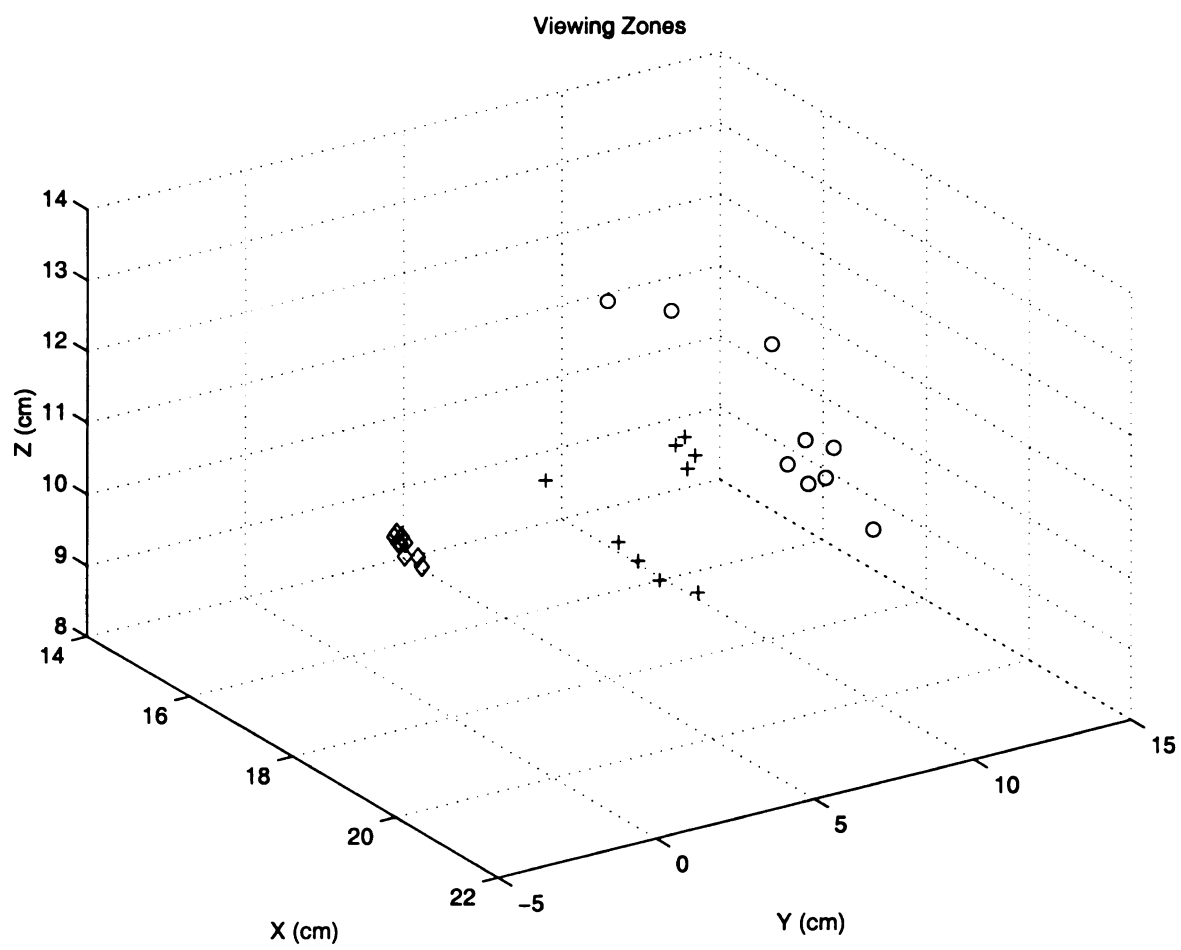


Table 5.1: Results for a single user from multiple viewing zones.

Observation	Zone 1 (cm)	Zone 2 (cm)	Zone 3 (cm)	Global (cm)
Mean Error	0.66	1.05	0.30	0.67
Max Error	1.20	1.75	0.50	1.75

Table 5.2: Average errors for different users from multiple viewing zones.

Zone 1 (cm)	Zone 2 (cm)	Zone 3 (cm)	Global (cm)
0.66	1.05	0.30	0.67
0.71	0.97	0.64	0.77
0.38	0.79	1.19	0.79
1.21	1.52	1.08	1.27
0.77	0.71	0.85	0.78
1.21	1.68	0.52	1.14
1.16	1.63	0.81	1.20
1.00	0.72	0.82	0.85

errors and maximum errors for eight different user sessions. For these eight sessions there were six users. The errors shown in rows 1 and 2 and rows 3 and 4 are from the same users during two different sessions.

From the tables it can be seen that the overall average system accuracy is 0.93cm. The worst error observed was 4.73cm, but this was caused by the user moving outside the effective range of the tracker. If that point is treated as an outlier then the worst observed error was 3.32cm.

The results for the video-based system are shown in Figure 5.6. In this plot, the + symbols represent the actual locations of the points provided by the validation modules and the × symbols show the observed locations. The circles provide visual feedback of the errors at each location.

Table 5.3: Maximum errors for different users from multiple viewing zones.

Zone 1 (cm)	Zone 2 (cm)	Zone 3 (cm)	Global (cm)
1.20	1.75	0.50	1.75
0.90	1.25	0.90	1.25
0.82	1.26	2.02	2.02
3.23	2.39	1.85	3.23
1.72	1.13	1.21	1.72
2.35	2.76	1.22	2.76
1.92	2.54	1.15	2.54
4.73	3.19	3.02	4.73

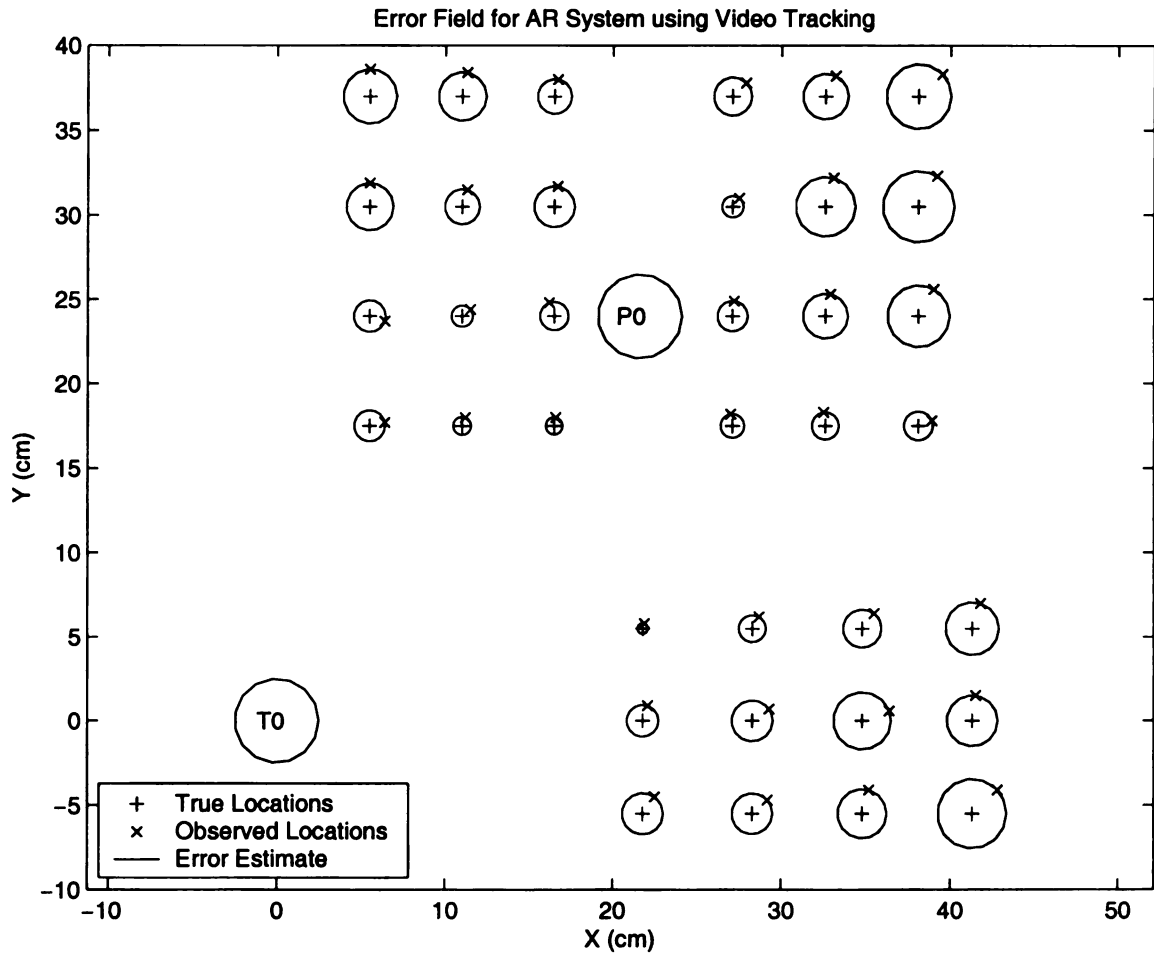


Figure 5.6: Error fields for an AR system which uses a infrared video tracking system. *P0* indicates the calibration origin and *T0* indicates the tracker origin.

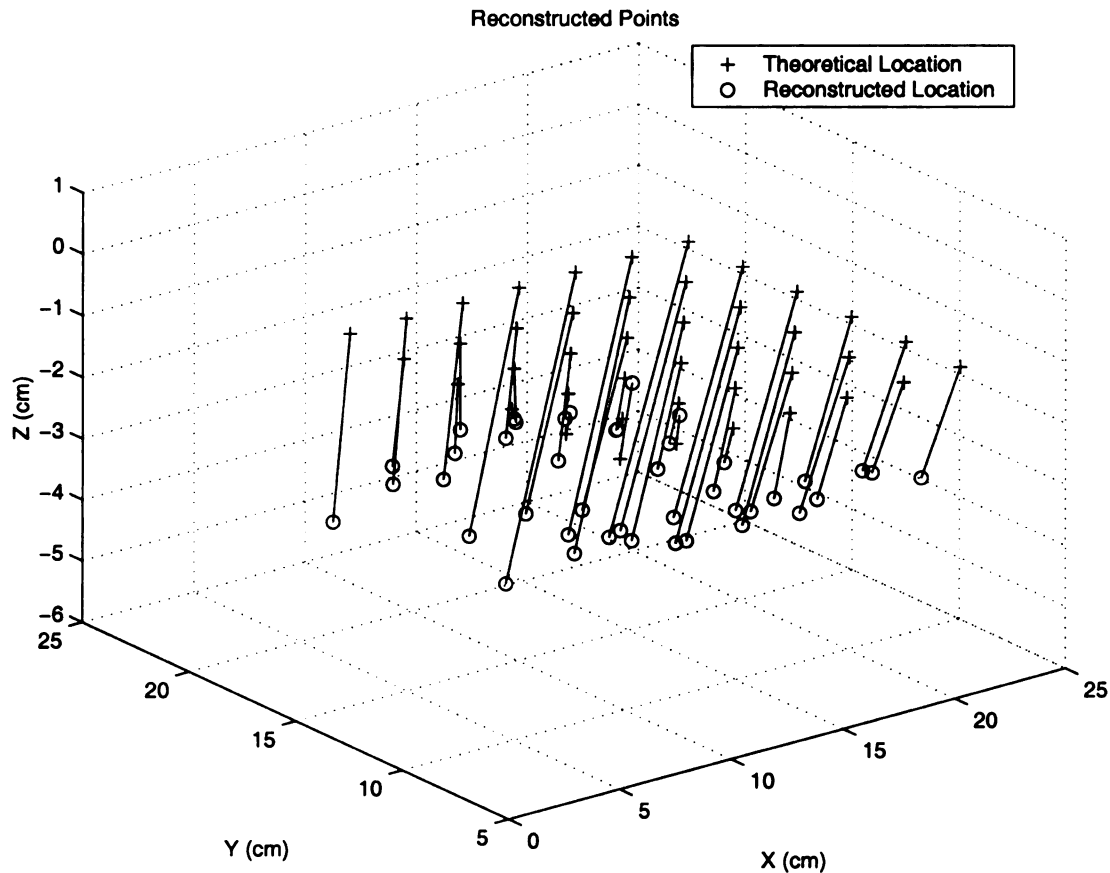


Figure 5.7: Reconstructed locations

5.4 Analysis of Structured Errors

Given multiple observations of a point from different viewpoints it is possible to estimate its 3-dimensional location. If all of the points from the evaluation task are estimated they can be looked at collectively as an error field. Figure 5.7 shows an example of such an error field. In this plot, the plus symbols represent the theoretical locations of the points and the circles represent the reconstructed locations. Notice that the reconstructed points are roughly coplanar. This implies that a correction can be performed upon the calibration.

Given two or more image plane observations of the same point, and the render matrix for each observation its 3-dimensional location $\mathbf{p}$ can be estimated by solving

$$\mathbf{A}\mathbf{p} = \mathbf{b}, \quad (5.1)$$

where

$$\mathbf{A} = \begin{bmatrix} g_{11,1} & g_{12,1} & g_{12,1} \\ g_{21,1} & g_{22,1} & g_{22,1} \\ g_{11,2} & g_{12,2} & g_{12,2} \\ g_{21,2} & g_{22,2} & g_{22,2} \\ \vdots & \vdots & \vdots \\ g_{11,n} & g_{12,n} & g_{12,n} \\ g_{21,n} & g_{22,n} & g_{22,n} \end{bmatrix}, \quad (5.2)$$

and

$$\mathbf{b} = \begin{bmatrix} u_1(g_{31,1} + g_{32,1} + g_{33,1} + g_{34,1}) - g_{14,1} \\ v_1(g_{31,1} + g_{32,1} + g_{33,1} + g_{34,1}) - g_{14,1} \\ u_1(g_{31,2} + g_{32,2} + g_{33,2} + g_{34,2}) - g_{14,2} \\ v_1(g_{31,2} + g_{32,2} + g_{33,2} + g_{34,2}) - g_{14,2} \\ \vdots \\ u_n(g_{31,n} + g_{32,n} + g_{33,n} + g_{34,n}) - g_{14,n} \\ v_n(g_{31,n} + g_{32,n} + g_{33,n} + g_{34,n}) - g_{14,n} \end{bmatrix}. \quad (5.3)$$

From the solutions it can be seen that there is some structure to the error. Using this structure it may be possible to make adjustments to the various calibrations in the system. This is left for future work.

5.5 Conclusions

From the results, it is now possible to describe applications which can be built using the AR system which was tested. Given that the system tested has an average error on the order of 1cm and a maximum error of around 3cm, it would be possible to build several useful applications with it. In fact, it would be possible to implement most of the applications listed in Chapter 1. Games and entertainment, *e.g.*, Hunt the Wumpus, Tic-Tac-Toe, are quite practical, provided the game objects are large enough (6cm or larger) to correct for interaction errors.

A maintenance or inspection task could also be implemented, again, given that the objects that need to be worked with are larger than the system tolerance. For instance, an application to look for missing components on a circuit board would be feasible, an application which required the user to touch a probe to a contact would not. This sort of application would require millimeter accuracy.

Finally, surgery would be completely out of the question. In such an application, the accuracy demands would be sub-millimeter. Deeper study of optical see-through calibration methods and more accurate tracking systems will be needed to achieve this goal. Also, robust real-time registration systems for the medical data must be integrated into the system. The author will be exploring the effects of a better tracking system as future work.

Chapter 6

Summary and Conclusions

This thesis presents a general method to objectively evaluate an optical see-through augmented reality system. This method differs from previous methods in three ways. First, it allows a user to measure the error of the system while it is running. Second, it produces quantitative measurement of the error in registration of the real and the virtual. Finally, this error estimate includes the user as part of the system. With this method it is possible to estimate the accuracy limits of an application and detect structured errors in the tracking system. The method also gives insight which can be used to design new calibration algorithms.

Several conclusions can be drawn from the experimental results. First, an AR system must be evaluated as a whole. While the errors of individual system components are both useful and meaningful to an implementor, they would be of little value to a maintenance technician using an AR system to perform repairs. Users need concise, simple feedback on the current system calibration accuracy.

Second, during an evaluation of an AR system, the user must be factored in.

Since all of the calibration procedures for the system components involve a manual step, there is a possibility that the user can introduce errors into the system. This is especially true if the user is ignorant of actions which may cause errors, such as facial muscle contractions during the calibration, *e.g.*, talking, raising the brow, etc. Also, simplicity is a must for any calibration algorithm. If the user has to perform some difficult action, it is likely they will inject an error into the system. Other factors may cause errors during calibration and evaluation. For instance, poor lighting which can make alignment of the stylus to the observed point less accurate since the user has difficulty seeing the stylus tip through the display.

Third, since the observed errors are dependent on the viewpoint, the verification should be performed from all zones that the user intends to work in. Also, multiple planes may be evaluated throughout the workspace. From multiple views of the same plane the 3-dimensional error can be estimated. This estimation uses a technique from computer vision known as point reconstruction [7, 16]. Figure 5.7 shows the reconstructed locations of the points from a user. It may be possible to make corrections to a calibration using this information.

APPENDICES

Appendix A

User Error Field Plots

This appendix contains the error field plots from eight trials of the magnetically-tracked optical see-through system. These six users had varying amounts of experience with the system. Figures A.1 and A.2 and Figures A.3 and A.4 were generated from the two primary investigators. The rest were volunteers. Careful study of these figures shows that the accuracy limit for the system is between 1 and 2 centimeters for a typical user.

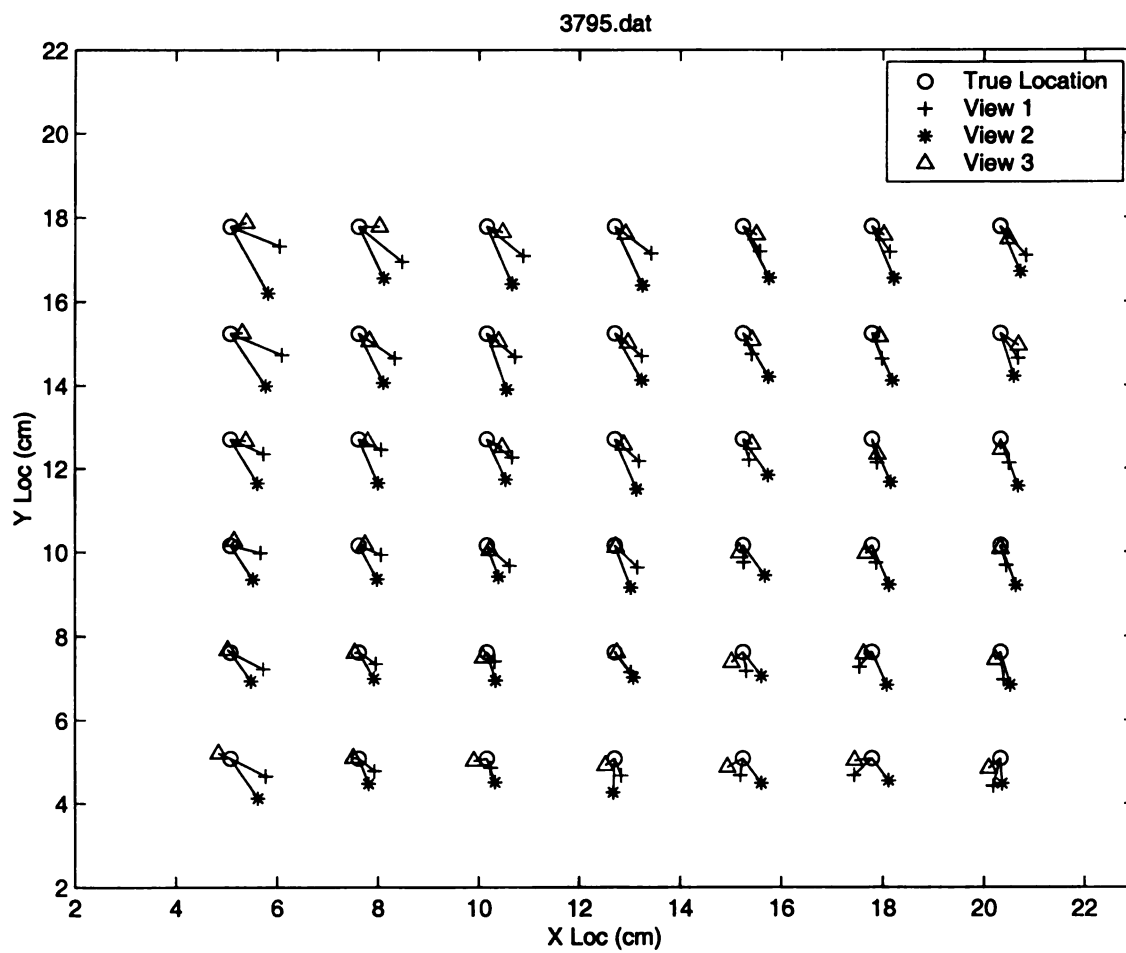


Figure A.1: Three-view errors from a user.

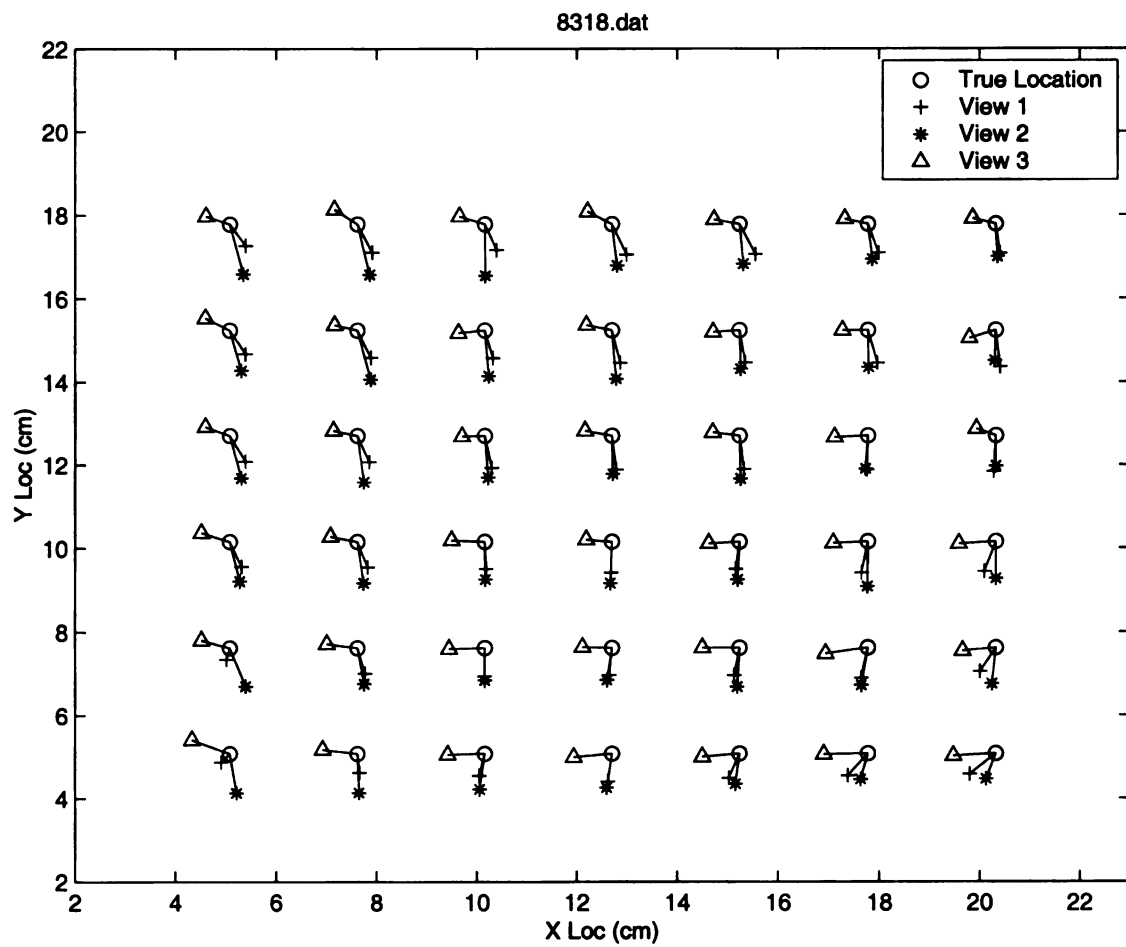


Figure A.2: Three-view errors from a user.

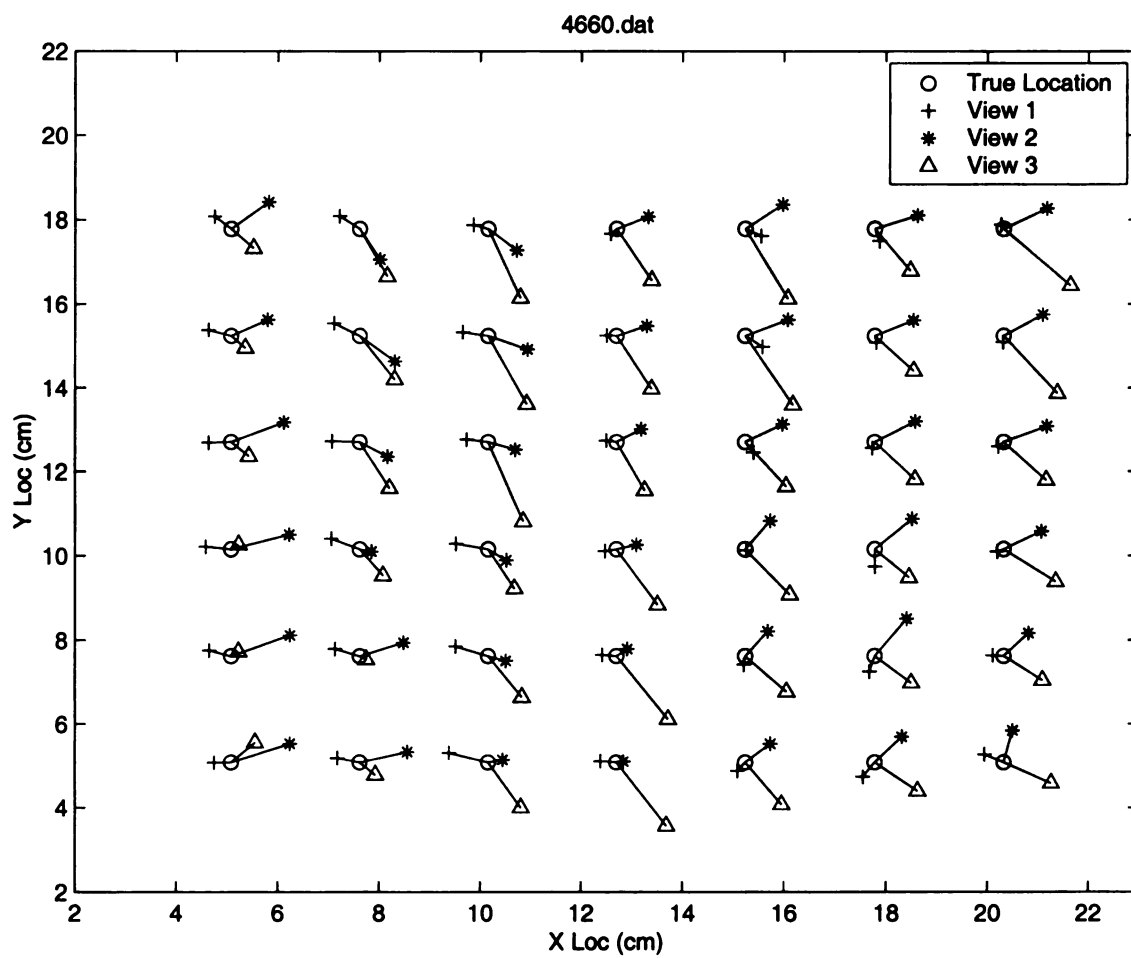


Figure A.3: Three-view errors from a user.

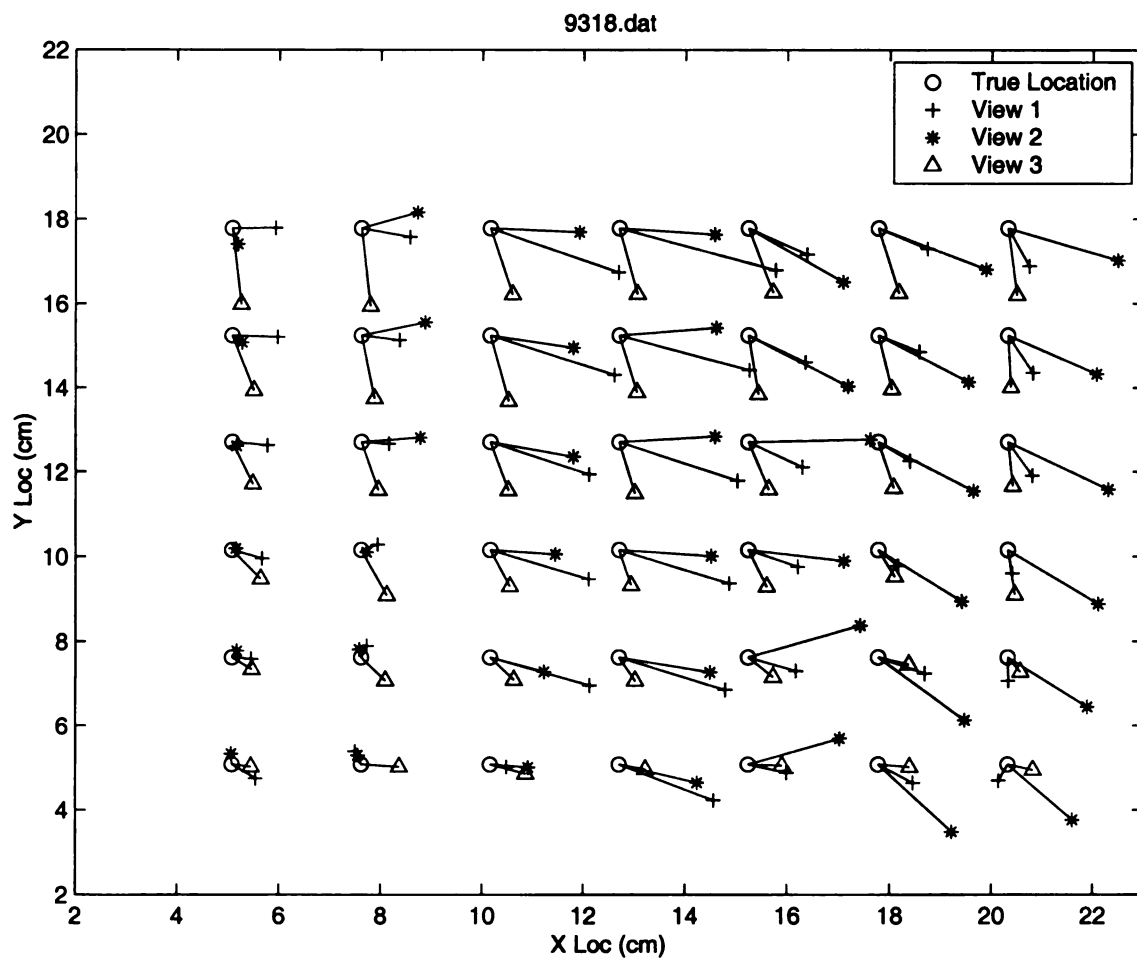


Figure A.4: Three-view errors from a user.

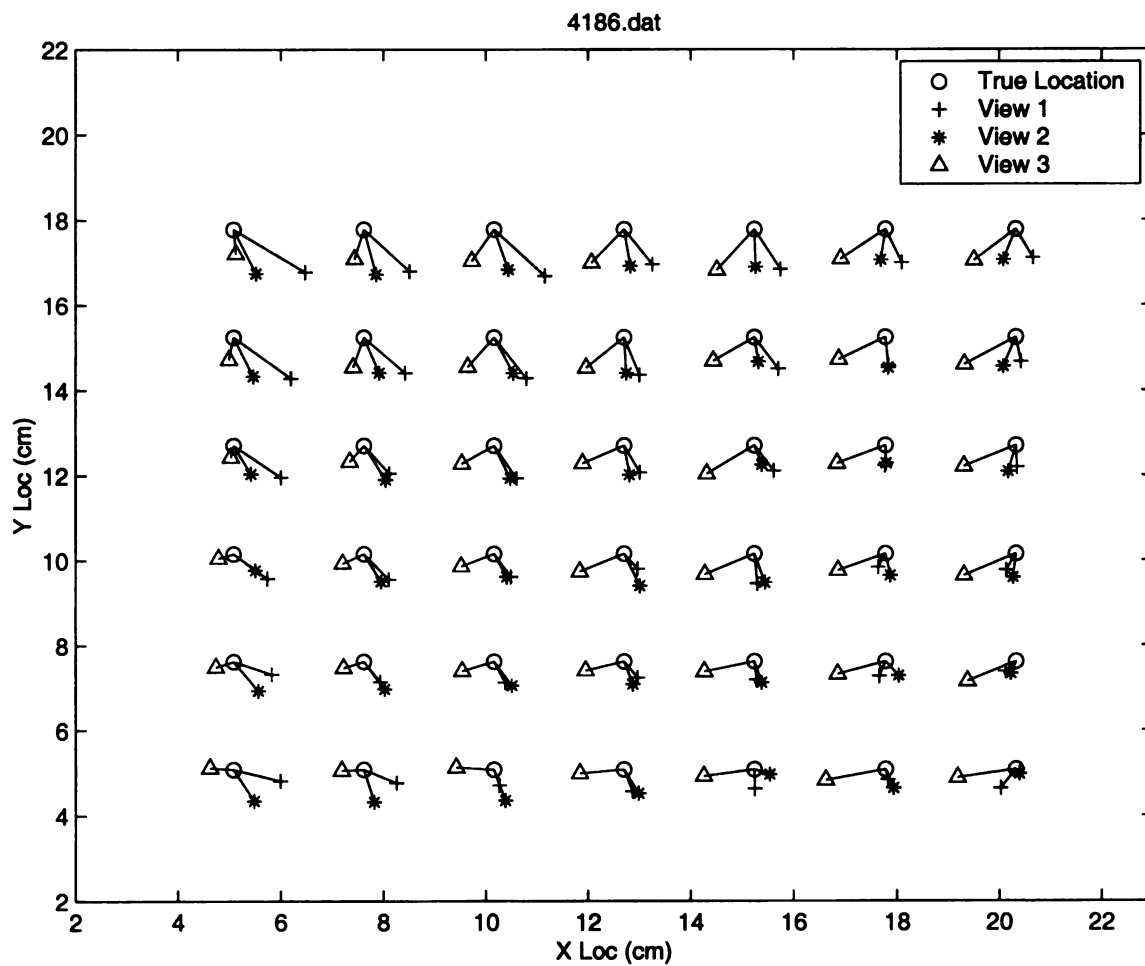


Figure A.5: Three-view errors from a user.

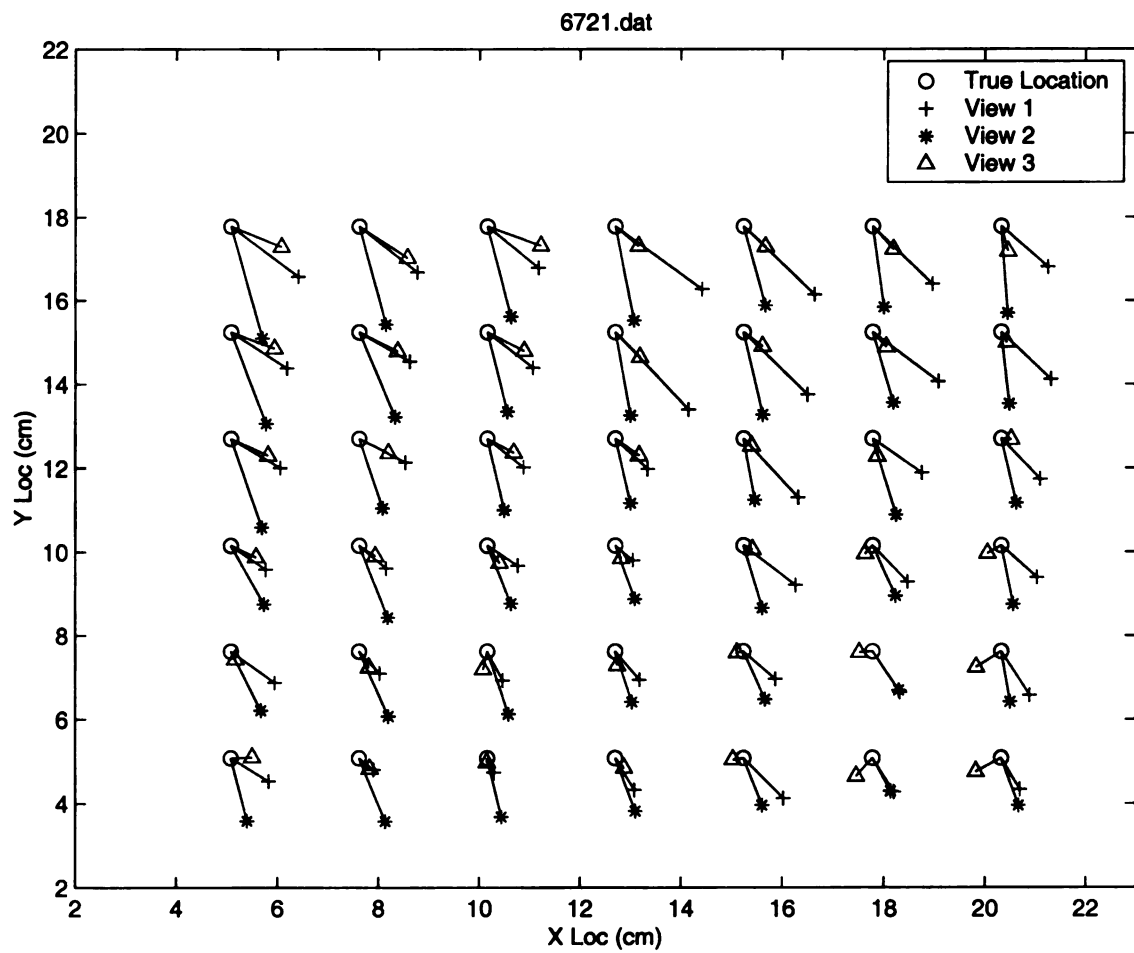


Figure A.6: Three-view errors from a user.

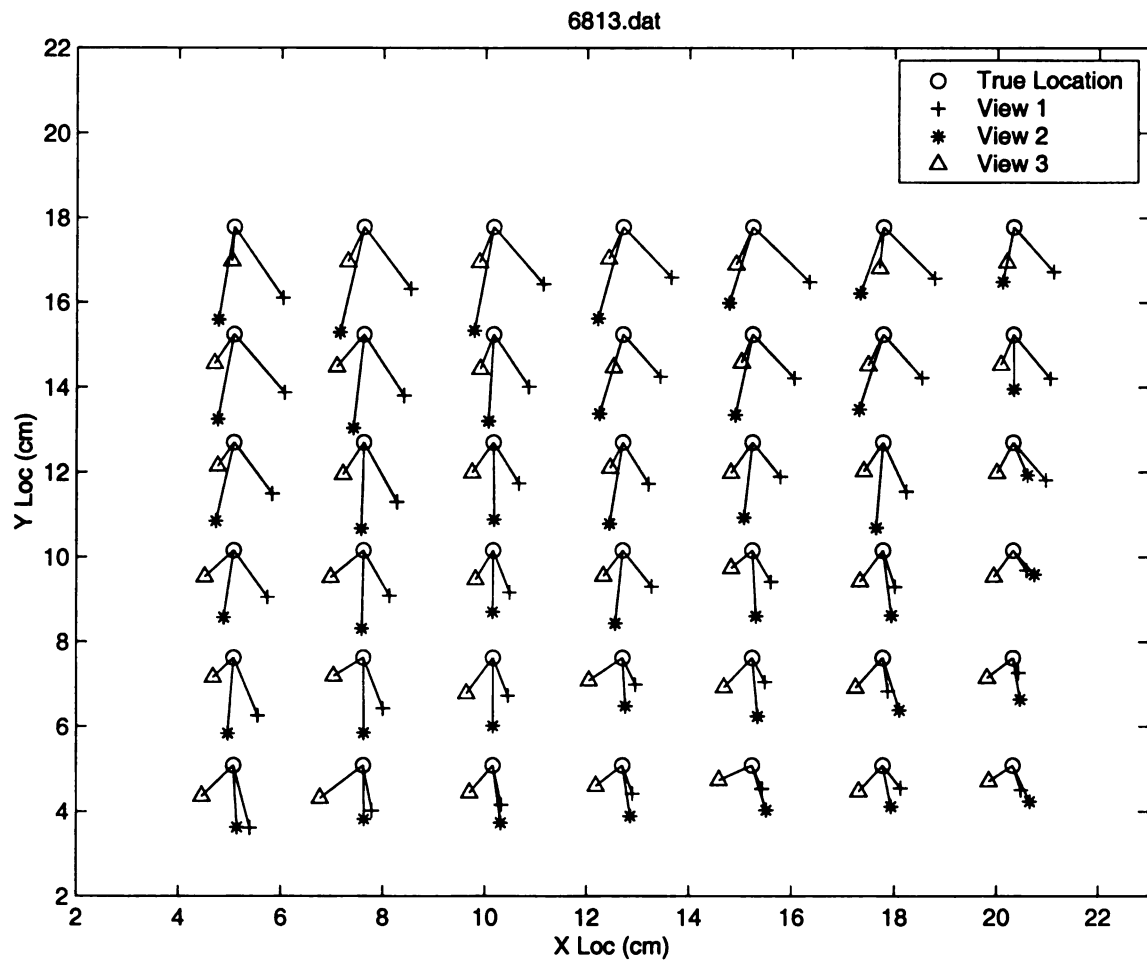


Figure A.7: Three-view errors from a user.

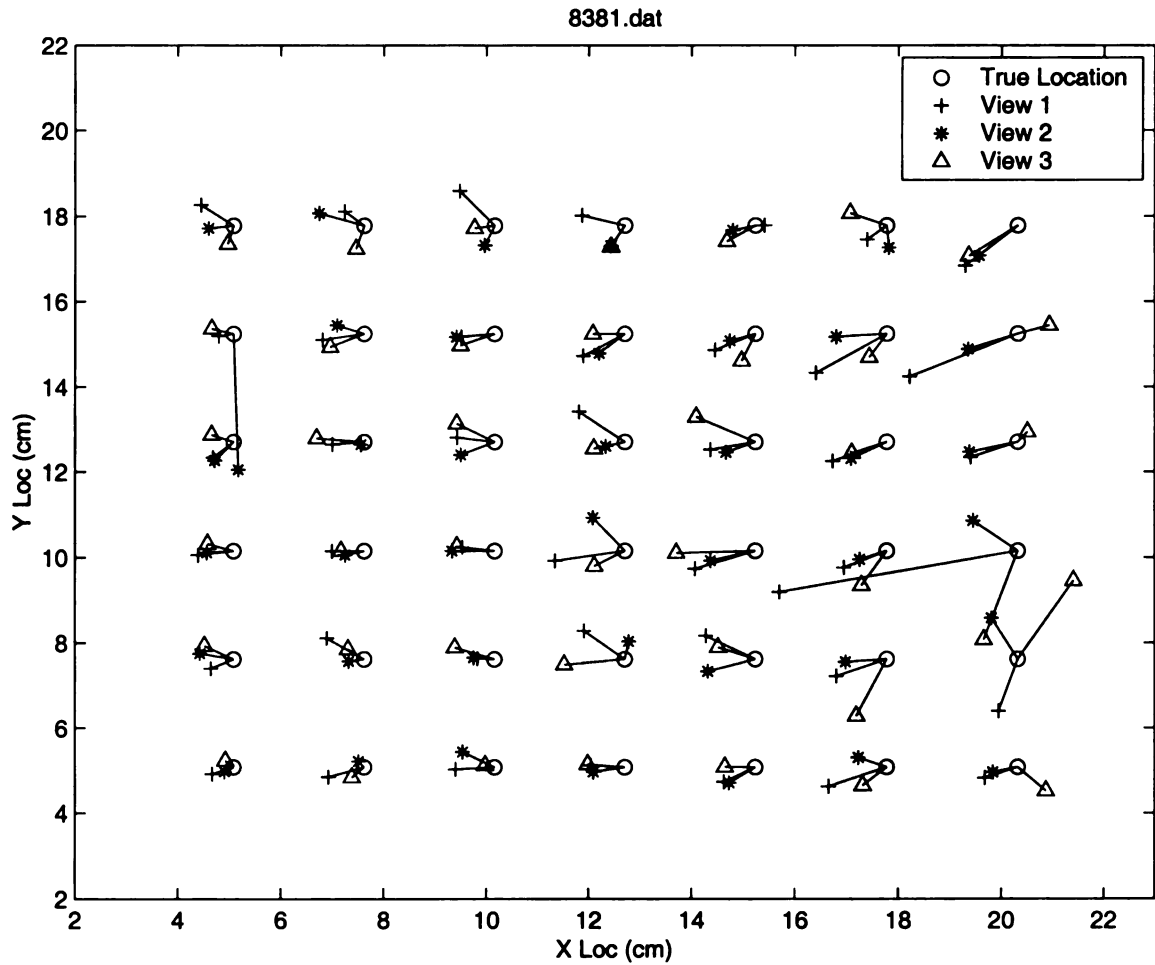


Figure A.8: Three-view errors from a user. The outliers in this figure are from the user moving outside the range of the tracker.

BIBLIOGRAPHY

Bibliography

- [1] R. Azuma. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385, August 1997.
- [2] R. Azuma and G. Bishop. Improving static and dynamic registration in an optical see-through display. *Computer Graphics*, pages 194–204, July 1994.
- [3] M. Bajura, H. Fuchs, and Ohbuchi. Merging virtual objects with the real world: Seeing ultrasound imagery within the patient. *Computer Graphics*, pages 203–210, July 1992.
- [4] M. Bajura and U. Neumann. Dynamic registration correction in augmented-reality systems. In *Proceedings of the Virtual Reality Annual International Symposium (VRAIS '95)*, pages 189–196, 1995.
- [5] W. Birkfellner, K. Huber, F. Watzinger, M. Figl, F. Wanschitz, R. Hanel, D. Rafolt, R. Ewers, and H. Bergmann. A see-through hmd for computer-aided surgery. In *3rd International Workshop on Augmented Reality (IWAR 2000)*, pages 54–59, Munich, Germany, October 2000.
- [6] A. Butz, T. Hollerer, S. Feiner, B. MacIntyre, and C. Beshers. Enveloping users and components in a collaborative 3d augmented reality. In *2nd International Workshop on Augmented Reality (IWAR '99)*, pages 35–44, San Francisco, CA, October 1999.
- [7] O. Faugeras. *Three-dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, Cambridge, MA, 1993.
- [8] S. Feiner, B. MacIntyre, and D. Seligmann. Knowledge-based augmented reality. *Communications of the ACM*, 36(2):53–62, 1993.
- [9] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [10] H. Fuchs and J. P. Rolland. Optical versus video see-through head-mounted displays in medical visualization. *Presence*, 9(3):287–309, 2000.

- [11] Y. Genc, N. Navab, M. Tuceryan, and E. McGarrity. Evaluation of optical see-through systems. Technical Report SCR-01-TR-688, Siemens Corporate Research, January 2001.
- [12] Y. Genc, F. Sauer, F. Wenzel, M. Tuceryan, and N. Navab. Stereo optical see-through HMD calibration: A one step method validated via a video see-through system. In *International Symposium for Augmented Reality*, pages 165–174, Munich, Germany, October 2000.
- [13] W.E.L. Grimson, G. J. Ettinger, S.J. White, P.L. Gleason, T. Lozano-Perez, III W. M. Wells, and R. Kikinis. Evaluating and validating an automated registration system for enhanced reality visualization in surgery. In *Proc. of the IEEE Conference on Computer Vision, Virtual Reality and Robotics in Medicine (CVRMed'95)*, pages 3–12, 1995.
- [14] R. Holloway. *An Analysis of Registration Errors in a See-Through Head-Mounted Display System for Craniofacial Surgery Planning*. PhD thesis, University of North Carolina at Chapel Hill, 1994.
- [15] R. L. Holloway. Registration error analysis for augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):413–432, August 1997.
- [16] Ramesh Jain, Rangachar Kasturi, and Brian G. Schunck. *Machine Vision*. McGraw-Hill, New York, 1995.
- [17] A. Janin, D. Mizell, and T. Caudell. Calibration of head-mounted displays for augmented reality applications. In *Proc. of the Virtual Reality Annual International Symposium (VRAIS'93)*, pages 246–255, 1993.
- [18] H. Kato and M. Billinghurst. Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In *Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality '99*, pages 85–94, San Francisco, CA, October 20–21 1999.
- [19] V. Kindratenko. Calibration of electromagnetic tracking devices. *Virtual Reality: Research, Development, and Applications*, 4:139–150, 1999.
- [20] G. J. Klinker, K. H. Ahlers, D. E. Breen, P.-Y. Chevalier, C. Crampton, D. S. Greer, D. Koller, A. Kramer, E. Rose, M. Tuceryan, and R. T. Whitaker. Confluence of computer vision and interactive graphics for augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):433–451, August 1997.
- [21] K. N. Kutulakos and J. R. Vallino. Affine object representations for calibration-free augmented reality. In *Proceedings of the IEEE Virtual Reality Annual International Symposium*, 1996.
- [22] W. Lorensen, H. Cline, C. Nafis, R. Kikinis, D. Altobelli, and L. Gleason. Enhancing reality in the operating room. In *Proceedings of the IEEE Conference on Visualization*, pages 410–415, 1993.

- [23] D. G. Lowe. Robust model-based motion tracking through the integration of search and estimation. *International Journal of Computer Vision*, 8(2):113–122, August 1992.
- [24] E. McGarrity and C. B. Owen. Aris: Augmented reality inspection system. Technical report, Michigan State University Manufacturers Research Consortium, East Lansing, Michigan, 2000.
- [25] E. McGarrity and M. Tuceryan. A method for calibrating see-through head-mounted displays for augmented reality. Technical Report TR-CIS-0499-13, Indiana University Purdue University Indianapolis, April 1999.
- [26] E. McGarrity, M. Tuceryan, C. Owen, Y. Genc, and N. Navab. Evaluation of optical see-through systems. In *EUROIMAGE International Conference on Augmented, Virtual Environments and Three Dimensional Imaging*, Mikanos, Greece, June 2001.
- [27] Erin McGarrity and Mihran Tuceryan. A method for calibrating see-through head-mounted displays for AR. In *2nd International Workshop on Augmented Reality (IWAR '99)*, pages 75–84, San Francisco, CA, October 1999.
- [28] J. P. Mellor. Real-time camera calibration for enhanced reality visualizations. In *Proc. of the IEEE Conference on Computer Vision Virtual Reality and Robotics in Medicine (CVRMed'95)*, pages 471–475, 1995.
- [29] U. Neumann and Y. Cho. A self-tracking augmented reality system. In *Proceedings of the ACM Symposium on Virtual Reality and Applications*, pages 109–115, 1996.
- [30] T. Oishi and S. Tachi. Methods to calibrate projection transformation parameters for see-through head-mounted displays. *Presence*, 5(1):122–135, 1996.
- [31] H. Tamura. What happens at the border between real and virtual worlds: The mr project and other research activities in japan. In *3rd International Workshop on Augmented Reality (IWAR 2000)*, pages xii–xv, Munich, Germany, October 2000.
- [32] M. Tuceryan, D. Greer, R. Whitaker, D. Breen, C. Crampton, E. Rose, and K. Ahlers. Calibration requirements and procedures for a monitor-based augmented reality system. *IEEE Transactions on Visualization and Computer Graphics*, 1(3):255–273, September 1995.
- [33] M. Tuceryan and N. Navab. Single point active alignment method (SPAAM) for optical see-through HMD calibration for AR. In *International Symposium for Augmented Reality*, pages 149–158, Munich, Germany, October 2000.
- [34] M. Uenohara and T. Kanade. Vision-based object registration for real-time image overlay. In *Proc. of the IEEE Conference on Computer Vision, Virtual Reality and Robotics in Medicine (CVRMed'95)*, pages 13–22, 1995.

- [35] Z. Zhang. A flexible new technique for camera calibration. Technical Report MSR-TR-98-71, Microsoft Research, March 1999.

MICHIGAN STATE UNIVERSITY LIBRARIES



3 1293 02112 5889