



136  
918  
THS

This is to certify that the

thesis entitled

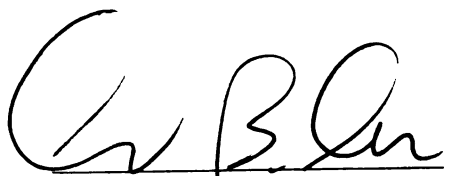
**EVALUATION OF REGISTRATION ERROR THROUGH  
OPTICAL MODELING OF THE DISPLAY EYE SYSTEM**

presented by

**Jonathan P. Babbage**

has been accepted towards fulfillment  
of the requirements for

M.S. degree in Computer Science and Engineering



Major professor

May 8, 2003

Date \_\_\_\_\_

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 |
|-----------------------|----------|----------|
| MAR 03 2009<br>060209 |          |          |
|                       |          |          |
|                       |          |          |
|                       |          |          |
|                       |          |          |
|                       |          |          |
|                       |          |          |
|                       |          |          |
|                       |          |          |
|                       |          |          |

EVALUATION OF REGISTRATION ERROR THROUGH OPTICAL  
MODELING OF THE DISPLAY EYE SYSTEM

By

Jonathan P. Babbage

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

2003

# **ABSTRACT**

## **EVALUATION OF REGISTRATION ERROR THROUGH OPTICAL MODELING OF THE DISPLAY EYE SYSTEM**

By

Jonathan P. Babbage

This thesis examines the system formed by the combination of the human eye and an optical see-through head mounted display (HMD) for Augmented Reality (AR) applications. It is critical that developers of AR systems know what expectations can be placed on the hardware of the system. What level of registration accuracy can we expect between the virtual content as displayed in an HMD and real content in the world? What errors are introduced? In the past, the error has been measured by comparing points displayed in the HMD with the location a user perceives those points. This gives a measurement of the total error in the system but does include other variables such as noise from the tracking system and human error in the pointing process. This thesis develops a model for the correspondence of points in an optical see-through HMD with points in the physical world as seen through the display, taking into account the parameters of the physical human-computer system and motion of the eye relative to the display. Typically the graphics that are generated in an augmented reality system are created using a pin hole camera model. The eye is much more complicated, thus this camera model may not be sufficient. A model of the human eye will be presented and used to analyze what happens to light that enters the eye, be it from a real world object or one presented in the HMD. Using this information we be able to model the registration of virtual content and real content in the HMD/eye system.

# TABLE OF CONTENTS

|                                                      |    |
|------------------------------------------------------|----|
| ABSTRACT                                             | ii |
| 1 INTRODUCTION                                       | 1  |
| 1.1 Real World - Virtual Alignment . . . . .         | 2  |
| 1.2 Related Works . . . . .                          | 5  |
| 1.3 Contributions of this Thesis . . . . .           | 6  |
| 1.4 Outline . . . . .                                | 7  |
| 2 COMPONENTS OF AN AUGMENTED REALITY SYSTEM          | 8  |
| 2.1 Hardware . . . . .                               | 8  |
| 2.1.1 Tracking System . . . . .                      | 11 |
| 2.1.2 System Processor . . . . .                     | 12 |
| 2.1.3 Head Mounted Display . . . . .                 | 12 |
| 2.2 Software . . . . .                               | 13 |
| 2.2.1 Calibration . . . . .                          | 14 |
| 2.2.2 Graphics Rendering . . . . .                   | 17 |
| 2.3 User . . . . .                                   | 18 |
| 3 MODEL OF THE DISPLAY-EYE SYSTEM                    | 19 |
| 3.1 Physiology of the Human Eye . . . . .            | 20 |
| 3.2 Values Used in Calculation . . . . .             | 21 |
| 3.3 Components of Head Mounted Display . . . . .     | 22 |
| 3.4 Measurements used for Model . . . . .            | 23 |
| 3.5 Considerations . . . . .                         | 24 |
| 3.6 Components as a System . . . . .                 | 24 |
| 4 METHODS                                            | 25 |
| 4.1 Overview of Methods . . . . .                    | 25 |
| 4.2 Mathematical Models . . . . .                    | 26 |
| 4.2.1 Sphere Intersection . . . . .                  | 27 |
| 4.2.2 Plane Intersection . . . . .                   | 27 |
| 4.2.3 Direction Computation . . . . .                | 28 |
| 4.2.4 Subsequent Surfaces . . . . .                  | 30 |
| 4.2.5 Pin-Hole Camera Model . . . . .                | 30 |
| 4.3 Modeling Behavior of the Entire System . . . . . | 32 |
| 4.3.1 Mapping Real World to Display . . . . .        | 33 |
| 4.3.2 Mapping Display to Real World . . . . .        | 33 |
| 4.4 Ray Selection . . . . .                          | 35 |
| 4.4.1 Searching . . . . .                            | 35 |

|       |                                                                    |    |
|-------|--------------------------------------------------------------------|----|
| 5     | RESULTS                                                            | 37 |
| 5.1   | Graphical Rendering Error . . . . .                                | 37 |
| 5.1.1 | Experiment Layout . . . . .                                        | 38 |
| 5.2   | Isolated Eye Movement . . . . .                                    | 42 |
| 5.3   | Movement of the Head Mounted Display Relative to the Eye . . . . . | 46 |
| 6     | DISCUSSION AND CONCLUSIONS                                         | 49 |
| 6.1   | Pin Cushion Effect . . . . .                                       | 49 |
| 6.2   | Handling Eye Movement . . . . .                                    | 51 |
| 6.3   | Relative Movement of the HMD . . . . .                             | 52 |
| 6.4   | Conclusion . . . . .                                               | 56 |
|       | BIBLIOGRAPHY                                                       | 57 |

## LIST OF TABLES

|     |                                                              |    |
|-----|--------------------------------------------------------------|----|
| 3.1 | Values for schematic eye . . . . .                           | 22 |
| 3.2 | Indexes of Refraction . . . . .                              | 22 |
| 3.3 | Values for Head Mounted Display . . . . .                    | 24 |
| 5.1 | Values of Real World Point and Perceived Rendering . . . . . | 42 |
| 5.2 | Values for Rotation of $13^\circ$ . . . . .                  | 44 |
| 5.3 | Values for Moving the Display . . . . .                      | 48 |
| 6.1 | Values for Radial Distortion Correction . . . . .            | 52 |

## LIST OF FIGURES

|     |                                                                   |    |
|-----|-------------------------------------------------------------------|----|
| 2.1 | Hardware components of and Optical See-Through AR System . . . .  | 11 |
| 2.2 | Formation of a Virtual Image . . . . .                            | 13 |
| 2.3 | Coordinate System Transformations . . . . .                       | 15 |
| 2.4 | Calibration of Coordinate System . . . . .                        | 16 |
| 3.1 | Horizontal Section of the Eye . . . . .                           | 20 |
| 3.2 | Surfaces and Media of the Eye (not to scale) . . . . .            | 21 |
| 3.3 | Optical Components of the Sony Glasstron . . . . .                | 22 |
| 3.4 | Possible Movement of the Sony Glasstron . . . . .                 | 23 |
| 4.1 | Refraction . . . . .                                              | 28 |
| 4.2 | Incident Angle . . . . .                                          | 29 |
| 4.3 | Resultant Angle . . . . .                                         | 30 |
| 4.4 | 2 Dimensional Pin Hole Camera . . . . .                           | 31 |
| 4.5 | 3 Dimensional Camera . . . . .                                    | 32 |
| 4.6 | Reflection in a Curved Mirror . . . . .                           | 33 |
| 4.7 | Initial Ray Selection for HMD to Real World Mapping . . . . .     | 34 |
| 4.8 | Search Vectors . . . . .                                          | 36 |
| 5.1 | Layout of Optical Components for Extra-Fovea Points . . . . .     | 38 |
| 5.2 | Points used to create optimal configuration . . . . .             | 39 |
| 5.3 | Comparison of Real World Points and Perceived Rendering . . . . . | 41 |
| 5.4 | Graph of Error Based on a Rotation of $13^\circ$ . . . . .        | 45 |
| 5.5 | Graph for Moving the Display . . . . .                            | 47 |
| 6.1 | Real World Pin Cushion Example . . . . .                          | 50 |
| 6.2 | Simulated Rotation of the Eye . . . . .                           | 53 |
| 6.3 | Simulated Display Movement . . . . .                              | 55 |

# CHAPTER 1

## INTRODUCTION

Augmented Reality (AR) is the blending of computer-generated virtual content with real content. While many paradigms are available in AR, this thesis focuses on the use of head mounted displays (HMD) that present a 2D projection screen overlaid over the field of view of a user, thereby enhancing a user's perception of the world. AR is related to Virtual Reality (VR), but instead of replacing the world with an alternative reality, the user is still allowed to perceive reality, only with computer enhancement. In effect, the world is blended with the virtual elements [28]. The approach eliminates the isolation typically experienced in a VR environment. AR also allows the use of real world context as a reference to virtual content. It is this alignment of the real world and virtual components that necessitates information being accurately displayed. This thesis examines the eye, HMD, and their contributions to system error.

Computers have been advancing at great speeds for many years. A simple example of this growth is seen in Moore's Law. Originally proposed by Gordon Moore in 1965, it proposed that the number of transistors on the integrated circuit would double every few years [30]. This prediction has been accurate for almost 40 years since it was proposed. The same progress has been seen in many different aspects of computing including memory capacity, processor speed, and network speed. During this time human interaction with the computer has not seen advancements at the same speed. In particular, the portal through which we communicate with a computer remains a tiny rectangle relative to the size of real desktops and work environments. The advent of AR is changing this method of interaction. It also introduces the problem

of aligning information with the real world. The quality of this alignment has been examined by Erin McGarrity [26]. This research only dealt with the overall error encountered by a user over a period of time.

An AR system has many components which contribute to the error measured by previous research. It was unknown how each part of the system added to the measured error. In order to increase the accuracy of the system it is necessary to know how different components are contributing to the error. Therefore, the eye and the display are isolated in order to determine their role in the system.

This thesis focuses on both the human eye and the HMD that presents images to the eye. Both of these are small objects and difficult to measure. In the case of the eye, measurement were obtained through medical literature. Once the measurements were known, models were created that allowed for the examination of behavior in different situations. These models predicted the behavior of the system, but in order to validate the results photos were taken. The images were compared to the results from the models to see if the models of the system mimicked the real behavior.

The simulations support the idea that the current HMD is capable of presenting much more accurate information to the user, and therefore it must be other aspects of the system that are producing a majority of the error. The optical properties of the system can be compensated for in software. The movement of the users eye does not change perceived locations drastically. Also, the movement of HMD does not produce a great deal of error, though the tilting of it may. Therefore if the tilting of the HMD is limited and the software corrections are made, the HMD can be considered accurate.

## **1.1 Real World - Virtual Alignment**

Consider the small task of going to a new movie theater. On a computer one could easily find a map as well as driving directions that would lead them to the theater. The user is now required to take this information from the computer, store it, and apply it

to the real world. Computer hardware now allows for new methods of displaying data and entering information. Augmented Reality can be defined as the combination of information from a computer with the real world [4].

This broad definition admits a wide range of activities into the Augmented Reality classification. Again consider the task of going to a new movie theater. The following are two different solutions to the problem that may be considered AR. In both examples the output from portable computer is presented to the user through a Head Mounted Display (HMD). The first application could be as simple as displaying text directions to the theater. This removes the requirement for a user to remember information for the duration of the drive. The second solution involves a more complicated combination of both real world information, user input, and virtual information. Add to the previous system a method for tracking a users pose and location. Pose can be defined as ones orientation in space. Now one could present data such as distance until the next turn, estimated time until arrival, graphics that would direct the user to turn on the appropriate street, and an error message if the user is not headed in the right direction. The idea of using pose information coupled with a HMD it create a more inversive display can be traced back to the ideas of Sutherland [3, 2, 39, 40]. The real world advantages as such an immersive displayed have been researched by Tang [41].

These two examples demonstrate the idea that AR applications can have varying degrees of immersion. The range of what can be considered AR is broad, with many applications lying at different points along the spectrum. It is for this reason that it is useful to consider AR as being part of a continuum that occupies the space between a real environment and a completely virtual environment [28]. There are interesting ideas that the concept of a continuum introduces. It allows one to discuss the location of an application in the spectrum. A second concept that may be introduced is the idea of sliding a program in either direction along the spectrum. One could consider

how the application would change if the user was made more aware of the real world or if it were more obscured. The system discussed in this thesis works by presenting virtual content in a display such that the content overlays the visual field. A user sees reality with virtual elements added. Display technologies exist that present this overlay as an addition to the light viewed at each pixel or as a replacement [33]. Displays present content to each eye, thereby allowing for perception of the augmented content as 3D, fully registered with reality. Proper registration requires the placement of the graphics content at exactly the correct location in the display so as to be along the ray from the pupil to the corresponding physical location. Hence, a major element of any AR system is a method for ensuring the correct registration between the virtual content and reality.

Other AR paradigms, such as augmented imagery, the augmentation of still and video images, allow for the automation of the registration process. The final image as seen by the user is available to the computer for analysis. This is also true for video see-through head mounted displays. In this class of display the user perceives the world through a camera and an opaque display much as has been traditionally used for virtual reality systems. Again, the combined image as perceived by the user is available for analysis, allowing for automatic registration mechanisms and direct measurement of registration performance.

This process is complicated in an optical see-through display because the actual composition of the real and virtual content takes place on the retina of the user. An absolute capture of this composite image is not possible, so automatic methods for ensuring accurate registration based on image analysis are not possible. Hence, all existing approaches to optical see-through HMD calibration and registration require human interaction. This thesis seeks to determine what performance is actually possible in the eye/display system and how to achieve the best possible performance.

In the second example given above there is a great deal of error that a user could

accept. If the system were off by a measurement as drastic as a few feet, the graphics may instruct the user to turn before reaching the correct location. The user in the system, a driver, could possibly realize this error as being directed to turn onto a side walk or into a street sign. In this case the user would still be able to identify and turn onto the correct street quite easily. This is not the case for all possible applications. If the application was moved along the AR continuum in the direction of being completely virtual, it would not be so easy for the user to correct this error. It is possible the system would block out the user's ability to see the road in the real world. It is also possible the information being presented does not correspond to some real world object or the user is not able to observe the real object. The user would have no reference to the real world and thus not be able to correct the error. This is the case in research done at University of North Carolina. They designed a system to present medical information on a patient [23, 13]. In this case an error of a centimeter could be very drastic if it were used in a surgical setting [24].

Error is introduced into the system from two main sources. The first is inaccurate tracking. If the pose information from the tracking system is off by a degree, this could result in an error of a centimeter at arms length. The other source is inaccurate rendering of the graphics. Graphics rendering systems often use a pin hole camera model to create images. This model will be discussed later, but it is enough to know that the human eye is considerably more complex than a pin hole camera. Complexities of the eye include the fact that it is capable of changing shape, has multiple optical surfaces, and is able to move to focus on different objects.

## 1.2 Related Works

The first step in researching the error in an AR system was moving from a qualitative to a quantitative measurement. This advancement was made by Erin McGarrity [26]. Previous to his work there was only a qualitative measure. A user could say the

system seemed to work well or that it had too much error. One needed to be able to make statement such as "the average error of the system is one centimeter."

In order to establish this measurement of error in the system, objects in the real world must be compared to virtual ones. This is done by defining a set of points in space. These points have real world locations that are used in the computation. Also the points are put into the virtual system. Then they can be rendered as virtual objects and displayed to the user. Now the user needs to input the location where they perceive the point. The user is able to see the graphics in the HMD. They then take a stylus and move it to the location they may see the point. The tracking system then converts this selected point into a real world location. This information, along with the original location, is used to compute the error. This process is repeated over a large number of points and produces a quantity that is the average error.

Other related work include Holloway's research in looking into all the different aspects that can contribute to system error [14, 15]. Other research has been done by Min that focuses specifically on stereo viewing [29]. Stereo viewing referees to the process of presenting a different image to each eye in order to mimic real world experiences. There has also been efforts to reduce the error in the eye/display system made by Barsky [5, 6]. This research aimed to reduce error through an updated rendering technique.

### **1.3 Contributions of this Thesis**

At this point an aggregate error is known, but is it not known how much each component contributes to the total. Therefore each component must be examined separate from the rest of the system in order to produce a measurement of the error that portion contributes. This thesis identifies the different types and magnitudes of error possible in the process of rendering and perceiving information. The contributions of this thesis are:

1. A model of the eye/display system and a method to simulate its behavior.
2. A mapping from real-world points to display points.
3. An inverse mapping from display points to real-world rays.
4. Experimental validation of the model and simulation methods.
5. A model for consequences of both eye and display movement during usage.

In order to accomplish this, the optical objects that are involved in both presenting graphics and perceiving them must be measured and modeled. This means that models for the HMD and the human eye are needed. The main result of the thesis is a measure of a quantity of error that is product of the interaction of the display and the eye. This error is a measurement that assumes perfect tracking and calibration.

## 1.4 Outline

This thesis is based on the results of computations done to model different situations that could arise in an AR system. Chapter 1 is an introduction to the key ideas in AR as well as a presentation of the motivation and contributions of the research. The Chapter 2 discusses the components that make up the particular AR system being modeled in this thesis as well as the current method of graphics rendering. Chapter 3 presents the mathematical models used in the computation of results. This chapter also contains measurement data from the components of the system. Chapter 4 details the steps involved in converting these models into a computational process that produces results. Chapter 5 contains the results from the different situations that were examined. The final chapter discusses and summarizes the results and possible future research.

## CHAPTER 2

# COMPONENTS OF AN AUGMENTED REALITY SYSTEM

There are a wide variety of augmented reality applications and each requires a unique and specific hardware configuration. The process of combining images from the real world with those generated by a computer is the primary defining factor in an AR system. This *composition* be accomplished in a variety of ways, although most solutions have three components in common. There must be hardware that is able obtain information about the world and present appropriate data properly registered with that information. Software must exist to connect hardware that senses the status of the system in the real world and display hardware that achieves the actual composition. The user is a critical component in an AR system, in as it is the user perception of augmented reality that is important in an AR application. The research and results presented in this thesis only deal with one particular AR system, optical see-through head-mounted displays with real-time presentation of registered computer graphics. However, it is possible to apply these methods to other systems, though the components may differ.

### 2.1 Hardware

As was stated earlier, all AR systems lie on a continuum of the “degree of Immersion. It is for this reason that the hardware used in such systems varies greatly. Also it complicates the view of what actually is needed to constitute an AR system. The simplest view is that every AR system for augmentation of vision is made up of three

hardware components. The first of these is the input device, often taking the form of a video camera or some other type of tracking device. The input from this device is handled by the system processor. Finally the output is presented in a display. The following AR systems demonstrate the vast differences in hardware.

One simple version of AR is *projective* AR. In this paradigm, a video projector is used to display information on some object within the user's field of view [35, 34]. The location of the projector is known, so it is possible to place graphics at a specific location in space. Systems often include a video camera which is capable of tracking the location of objects in space. A simple application could track the location of a piece of paper, and use the projector to place some image on the paper. If the user moves the paper, the image on it moves as well. The rendered graphics can be warped so as to appear mapped onto the surface, much as texture mapping is done in computer graphics applications. Projective AR systems have the advantage that the user is not independently instrumented in any way and does not require special display hardware. However, projective systems limit the space in which they can be used to specially prepared objects and surfaces.

An alternative approach presents information from a video camera to the user. This information may be seen on a computer monitor or displayed in an opaque head-mounted display. The user is effectively viewing the world through the eyes of a camera. This method, known as monitor-based AR when used on computer monitors or video see-through when an HMD is utilized, can use the data in the camera image to determine the pose and location of objects and render graphics to be added to the video image. This combined image is then presented to the user [22, 27, 18]. As this approach can be HMD-based, it can mobilize the display. As one moves around in a projective system, it is possible to see surfaces that would be obscured from the output of a stationary projector or set of projectors. Using a HMD this problem can be solved, and all locations the user can see can be augmented. An additional advantage of

monitor-based and video see-through approaches is that content in the original image can be occluded by virtual content. Projective systems are fundamentally additive in nature, adding the displayed content from the projectors to the illumination already present on the surface.

There are multiple methods for both displaying information to the user. Example methods for combining graphics include reflecting information off of a half-silvered mirror, actually combining real video and graphics before presenting them to the user, and writing directly on the retina of the eye with a laser.

Tracking technologies are used to provide an indication of the relative location of the user or camera in the real world. The tracking can be low resolution such as a Global Positioning System [36]. This technique uses information from satellites to establish some latitude and longitude location on the earth. It has the advantage of being able to track over the entire earth, but can have errors of a few meters and yields no information about pose. GPS is also occluded by buildings or terrain. There are other options that would add pose information but have a smaller range of operation. This could be done with inertial tracking, magnetic tracking, or with a camera and computer vision techniques. All of these options have advantage and disadvantages that must be considered when developing an AR application.

Three main hardware components are present in the AR system considered in this thesis. They are a magnetic tracker, system processor, and optical see-through HMD. These are the same components that were used in previously mentioned work regarding error measurement. It is for this reason that the total error for the system can be compared to the rendering error.

These three components are connected in series and information flows from the tracker, through the processor, and on to the HMD. The tracker reports information about pose and location of the user's head to the computer. The computer then feeds these values in the rendering process to produce graphics output. This output is sent

to the HMD that the user is wearing. The tracking allows for these graphics to be presented in reference to some location in the real world.

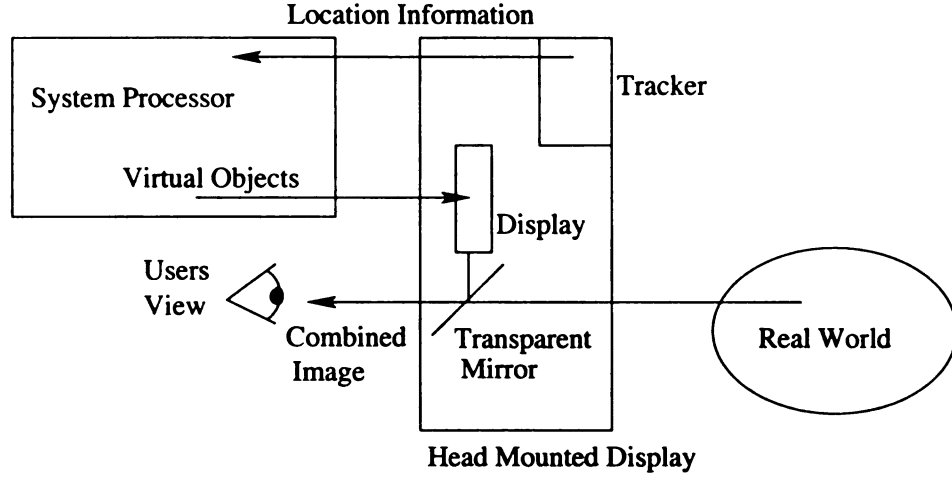


Figure 2.1: Hardware components of and Optical See-Through AR System

### 2.1.1 Tracking System

The tracking of a user in the AR system utilized in this thesis is accomplished using an Ascension Flock of Birds six degree-of-freedom magnetic tracking system. This hardware uses a transmitter to produce a magnetic field. Readings from up to 4 sensors that are placed in this field are then taken. The sensors are accurate within a range of 1.2 meters of the transmitter. The sensors have 6 degrees-of-freedom (6DOF) meaning that they can detect both location and pose (orientation). The optimal distance between the transmitter and sensors is 30.5cm. At this range the location accuracy is 1.8mm and the pose accuracy is 0.5°. The resolution for location is 0.5mm and for pose 0.1°. These values are the stated capabilities. In a lab setting there can be a large amount of interference in the magnetic field. Factors such as the presence of metallic objects as well as electronic interference from other computer components can decrease the accuracy of the tracking system [47].

Two sensors are used in this specific AR system. One sensor is mounted to a stylus

that is used to select points in space for calibration purposes and other interaction. The other sensor is mounted to the HMD. The data from this sensor is fed into the graphics rendering system. It is then used to align graphics with the real world. Also critical to the system is fixing the position of the transmitter in the world. If the transmitter were to move, it would affect the values reported for all the sensors, thus skewing the results.

### *2.1.2 System Processor*

The system processor has many different tasks in an AR system. It is the main connection between all the other pieces of hardware in the system. The system processor is responsible for providing the information from the tracker to the AR application. It runs the software that controls all parts of the AR system. The system processor also includes the hardware that produces the graphics for the HMD. The rendering in this system is done using OpenGL. OpenGL is a rendering technique that produces a 2 dimensional image from 3 dimensional models of objects. The specifics of the system processor such as manufacturer or clock speed are not included because they do not influence the rendered image or light from the real world. A different system processor given the same tracking information will produce the same graphical output. It is possible that the speed at which it does this may change but this is not a factor in the following computations. The only requirements are that the system be able to run OpenGL and connect to the tracking system.

### *2.1.3 Head Mounted Display*

The combination of real world and computer generated images is achieved using a method called Optical See-Through. This involves the use of a partially transparent mirror which reflects output from a display into the user's eye. Light from the real world is able to pass through this mirror and enter the eye. These two light sources,

the real world and the display, combine in the eye to form an augmented image for the user. The HMD used in this system is the Sony Glasstron. Since light passes through the display, it must be considered when modeling the optics of the system.

The Sony Glasstron is designed to simulate a large display, while being easily worn on the head. The display is seen as a  $762mm$  by  $571.5mm$  screen at a distance of  $1200mm$ . This is achieved using a concave mirror. A curved mirror has a focal length that is one half the radius of its curvature. If an object is placed within the focal distance of such a mirror, the reflected rays will not converge. If someone were to view these divergent rays, they would be able to see the object. The image they see is called a virtual image, and its location and size are determined by the location of the original object relative to the focal point of the mirror [21]. The rays involved are shown in Figure 2.2. The center of the radius of curvature is labeled  $C$  and the focal point  $F$ .

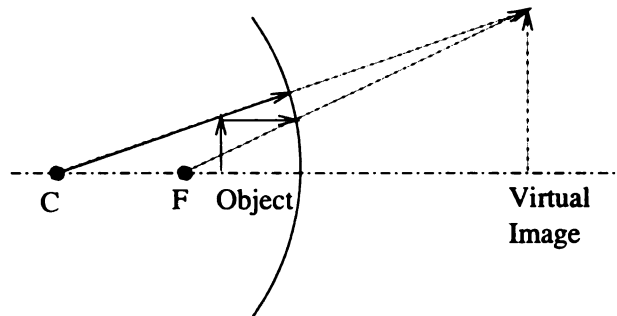


Figure 2.2: Formation of a Virtual Image

## 2.2 Software

Each AR application has three different tasks it must accomplish. It must first take readings from the AR system to establish a calibration. This is a process whereby the locations of objects and system components in the real world are reconciled with the locations of virtual elements to be used as augmentations. Once this is done it

must create a pipeline to the HMD using information from the tracker. This allows the application specific code to simply add virtual objects and the software will use the tracking information and present the graphics. The last task is different for each AR application, depending on the needs of the system. If the system is used to direct a user to a location, this task would include deciding when to instruct the user to turn. Since many AR applications share similar hardware components, calibration requirements, and tracking information, a system was created that allows for rapid generation of shell application. The shell application provides calibration capabilities and a pipeline to the HMD. This software is called ImageTclAR. It is a package created by Dr. Charles Owen and contributed to by researchers in the Michigan State University Metlab. A basic application was created using this software. This is the software used for this thesis.

### *2.2.1 Calibration*

The tracking hardware provides location and pose information relative to some point in the center of the transmitter and some point in the receiver, but the location of these points is not always known with great accuracy. All this data makes little sense in this format since the location of the transmitter may not correspond to the points that need to be tracked [9, 9, 45]. The goal of tracker calibration is to convert this information into something more meaningful. To start applying meaning to the data from the tracker, a main reference frame must be established. All coordinates can then be defined within this frame. This frame of reference is called the world coordinate system.

The idea of relative location, world coordinate system, and converting location data can be clarified by the following example. Consider a painting hanging on a wall. Any point in the painting could be described as being some distance from the top and left side of the frame. The same point in the painting has some location on

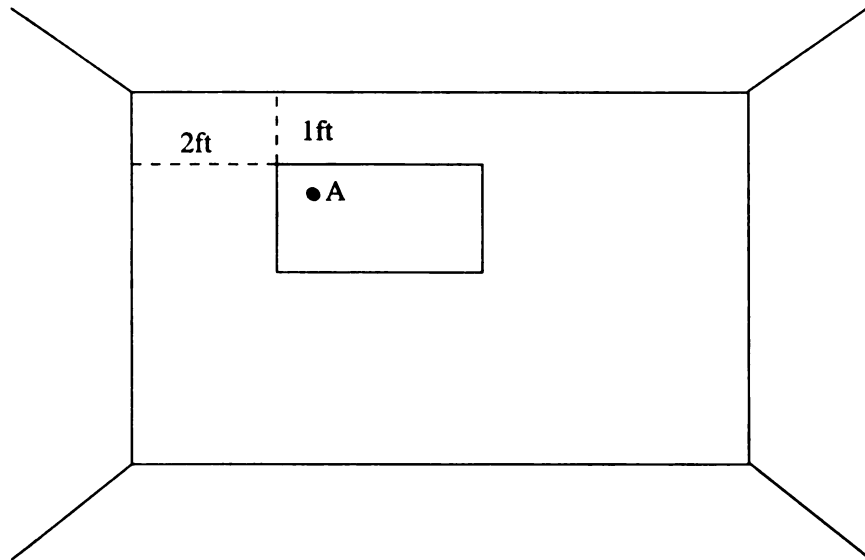


Figure 2.3: Coordinate System Transformations

the wall as well. This location could be a distance of five feet from the right wall and six from the floor. The data (5, 6) does not completely describe the location if the frame of reference is not known, since this could be the distance from the ceiling and left wall. Therefore when describing a location of a point, what frame of reference the measurements are relative to is very important.

The other idea that is demonstrated by the picture example is converting between coordinate systems. If the top left corner of the wall is defined to be the world coordinate system, the top left corner of the painting is (2, 1). Consider point *A* in the picture that is half a foot from the top of the frame and half a foot from the left. The location of this point is (2.5, 1.5) in reference to the world coordinate system. If the picture is moved to some new location on the wall, the location of it will have changed in reference to the wall but not in the painting. A conversion (transformation) can be created that will convert the location of *A* from the painting coordinates to the world coordinates. The conversion can be done by adding (.5, .5) to the location of the top left corner of the painting. The conversions used in calibrated rendering are more complicated, involving rotations or projections, but follow this

same basic concept.

In addition to the world coordinate system each component has its own coordinate system. The many different coordinate systems used are shown in Figure 2.4. The

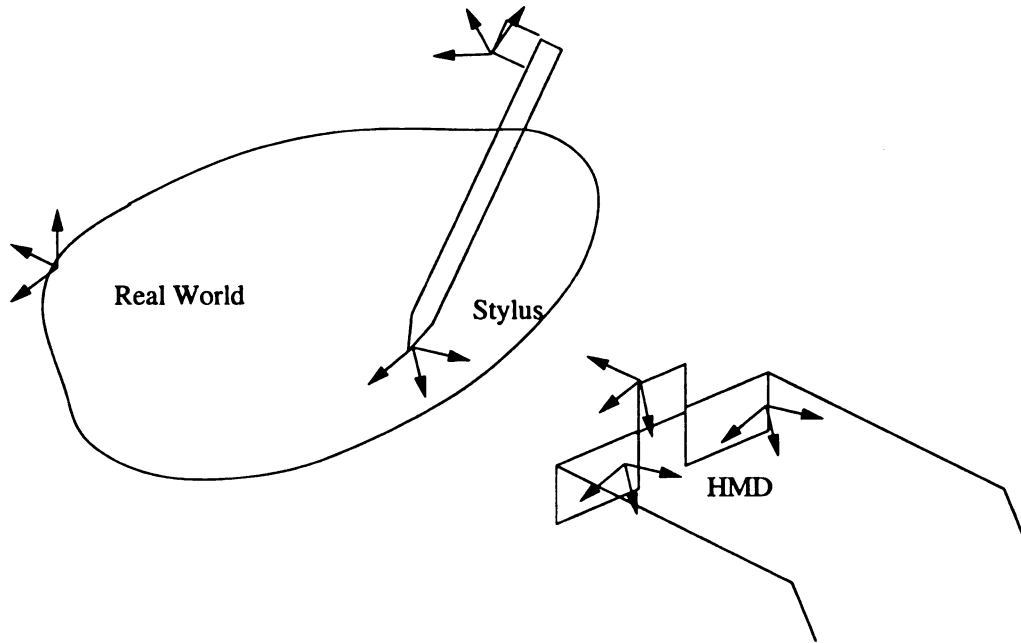


Figure 2.4: Calibration of Coordinate System

coordinate systems are the world, stylus receiver, stylus tip, HMD receiver, and both of the users eyes. Now a method is needed to convert between the coordinate systems. The conversion takes the form of a mathematical construct known as a transformation matrix. Using matrix multiplication, data in reference to one coordinate system may be easily converted to another system. There are three main conversions that must be established in order for the AR system to function. The calibration of the system takes place in a few steps since some transformations need to be established before the next one can be created.

The first step in calibration is establishing a means of selecting points in space. This is done using a stylus with a receiver mounted on it. (Not all AR systems utilize calibrated styli.) The data provided from the tracking system are the location and

pose of the receiver, which are different than the point of the stylus. At least six readings are taken while keeping the point of the stylus in one location and moving the stylus into different poses. This data is then used to produce a transformation from the location of the receiver to the point of the stylus. This allows for the selection of points in space using the tip of the stylus, which are then used to create a transformation from the tracker coordinate system to one in the real world. The stylus is used to select at least six known points on a rigid object in space. These are used to produce a second transformation that will convert locations from the transmitter coordinate system to the world coordinate system. The last step is the calibration of the HMD. An example method for this calibration has been developed by Tuceryan and Navab and is called SPAAM [25, 42]. This involves the user repeatedly aligning a point displayed in the HMD with a point in the real world. This is used to establish a location and properties of a camera that represents a pinhole camera model of the user's eye in combination with the optical see-through display. This process allows for the interaction with the real world using the stylus, the placing of virtual object in reference to a chosen world coordinate system, and the rendering of this entire virtual scene registered with the real world in the HMD.

### *2.2.2 Graphics Rendering*

All graphics that are to be aligned with or represent real world objects must be rendered from 3-D models. These graphics are rendered using OpenGL using calibration information from the SPAAM procedure. Objects are modeled using many polygons, which are made of a series of points, and added into a scene which OpenGL is to render. The information from the tracker mounted to the HMD dictates the location of the camera used for rendering. Thus, as a user moves their head around in space, the camera location is updated. Therefore, it appears to the user that the virtual objects are fixed in the real world since the graphics being drawn are always from the

viewpoint of a camera that corresponds to their eyes.

OpenGL uses a pin-hole camera model to render its graphics. Since this rendering is the critical source of error that is being examined in this thesis, the pin-hole model is discussed in section 4.2.5.

## 2.3 User

The final component of the AR system is the *user*. Human users introduce many variables in the processes of an AR system. Each user is different, having a differing interocular distance and placing the display at a slightly different location on the head. Some users have a great deal of experience with AR systems, some are complete novices. Some wear glasses. Some have eyes that have a different physiology than the norm of the population. Even if the same person were to calibrate and use the system at two different times, the location of the glasses can differ after being removed and remounted. All of these factors result in a different experience for each user, and possibly different experiences for the same user under different calibrations. This thesis assumes an average case and the results are computed from these assumptions. A more detailed computation is possible if measurements for each user are taken, but often this is difficult if not completely impractical. This is acceptable since the model can be changed to simulate some of the configurations of the system. This will provide more information about how changes to the system are reflected in the magnitude of error.

## CHAPTER 3

### MODEL OF THE DISPLAY-EYE SYSTEM

We are interested in determining exactly what performance is possible in an AR system utilizing an optical see-through head mounted display. In an optimum system, virtual augmentations would be precisely registered with sub-pixel accuracy with real elements. As an example, a wireframe of a box might be drawn over the physical box. In a perfect system, the wireframe would exactly follow the edges as seen from the viewpoint of the user of the system. The user would perceive the virtual lines as in the exact same location as the real box edges. An open question has been if this level of accuracy is even possible and, if so, what would be a mapping from points in the real world to equivalent points in the display. Clearly, if the eye does not move and all systems are static, such a mapping will exist with no error. But, what error will occur if the eye moves, the display moves, or if an approximation camera model, such as the pin-hole camera model, is used.

It is not possible to measure error directly in an optical see-through AR system. The location that light from a point source strikes the retina of the eye is not available (barring intrusive methods such as retinal photographs or heretofore unknown brain scanning technologies). In order to obtain the exact mapping from world points to display points, a mathematical model of the eye is needed. The eye is not the only component of the AR system that is involved in the perception of visual information by the user. Objects from the real world are viewed through the HMD. Also the overlaid virtual images pass through some components of the HMD. Thus models of the HMD and real-world objects are needed as well.

### 3.1 Physiology of the Human Eye

The human eye is a complicated system involving four refractive surfaces as well as a curved surface on which images are formed. Light strikes the cornea and travels through the anterior chamber. It then must pass through the opening of the iris and on into the main lens of the eye. Finally the light will strike the retina at the rear of the eye. All of these structures can be seen in Figure 3.1. In order to view different objects the eye is able to change configuration slightly. To focus on objects at different distances, the eye will change the shape of the lens. This is accomplished by contracting the ciliary muscles. The other way the eye can change its properties is by rotating inside the eye socket. There are six extra-ocular muscles that are used to rotate the eye around a fixed point [7].

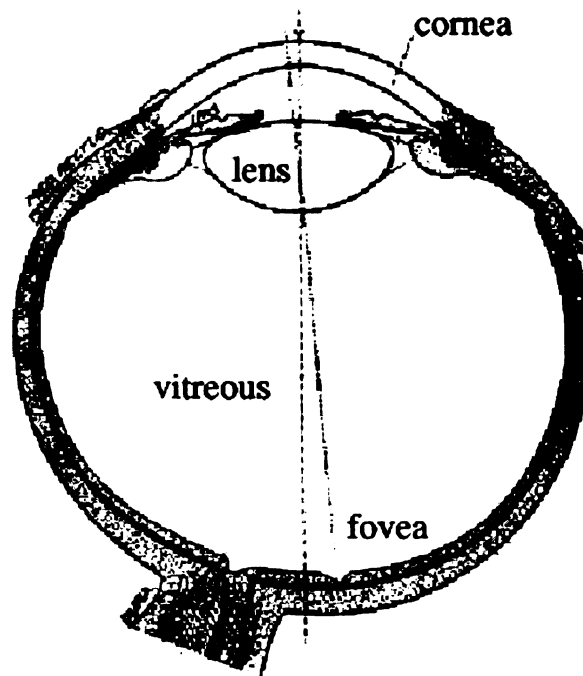


Figure 3.1: Horizontal Section of the Eye

In order to create a model of the eye it is necessary to know the shape of the different surfaces of the eye as well as the index of refraction of the materials that

make up the eye. The ideal situation would be if all the surfaces had shapes that followed a mathematical formula and the index of refraction were constant within each component. The eye differs from this ideal model in three ways. The surfaces of the eye are not exactly spherical. They tend to flatten out further away from the center, in particular the cornea does this more dramatically than other surfaces. Also the lens of the eye is not directly in line with the center of the eye. It is usually offset and on a slight angle. Lastly, the lens does not have a consistent index of refraction. The index of refraction is greater toward the center of the lens [21, 31]. All of these properties of the eye make modeling difficult.

### 3.2 Values Used in Calculation

In order to make a model possible some approximations are necessary. The model used in this thesis uses an approximation created by Gullstrand [31]. In this model the lens has been given a constant index of refraction based on an average. Also the surfaces are estimated to be completely spherical. The following distance measurements are all relative to the front of the eye.

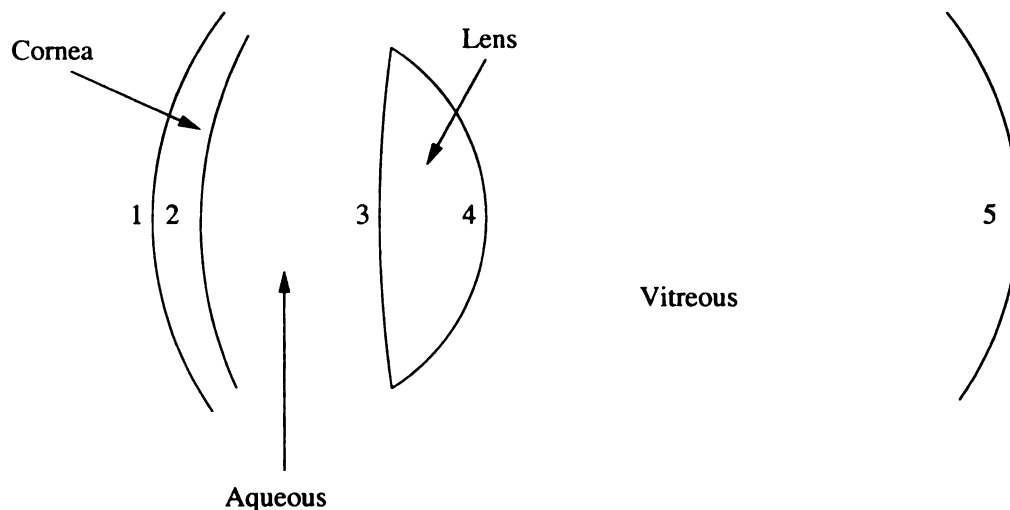


Figure 3.2: Surfaces and Media of the Eye (not to scale)

| Surface | Relative Location | Radius |
|---------|-------------------|--------|
| 1       | 7.7mm             | 7.7mm  |
| 2       | 7.3mm             | 6.8mm  |
| 3       | 13.6mm            | 10mm   |
| 4       | 1.2mm             | 6mm    |
| 5       | 13.3mm            | 11mm   |

Table 3.1: Values for schematic eye

|          |       |
|----------|-------|
| Cornea   | 1.376 |
| Aqueous  | 1.336 |
| Lens     | 1.41  |
| Vitreous | 1.336 |

Table 3.2: Indexes of Refraction

### 3.3 Components of Head Mounted Display

The display used in the particular AR system being modeled is a Sony Glasstron. It displays an image that appears to be approximately 1200mm from the user. This is accomplished with the use of curved half silvered mirror and a few other simple optical objects. The graphics are presented on a small LDC display in the top of the

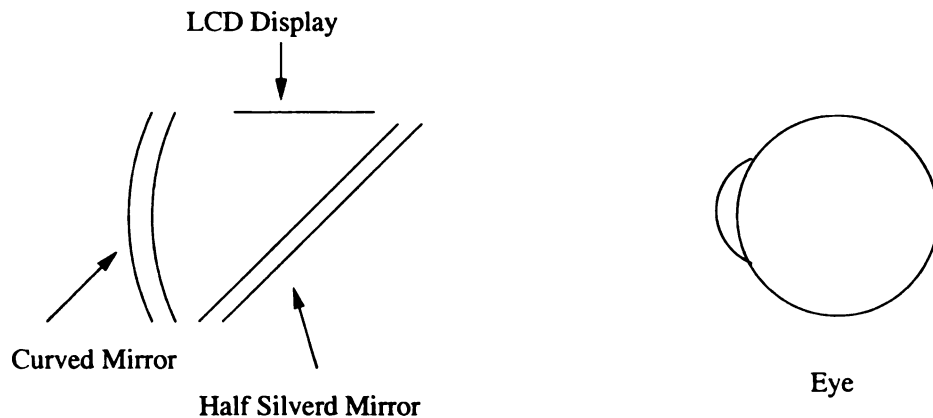


Figure 3.3: Optical Components of the Sony Glasstron

Glasstron. The light from the display strikes a mirror positioned at 45°. This reflects the light into the curved mirror, which then reflects back through the angled mirror

and into the eye of the user. The display is simplified to create the model. This is done by changing the location of the LCD in the model. It is moved to be directly in front of the curved half-silvered mirror. Optically this is the same as the real display but without reflecting light off the  $45^\circ$  mirror to direct it into the primary mirror. The HMD is placed on the users with a pad contacting the forehead. There

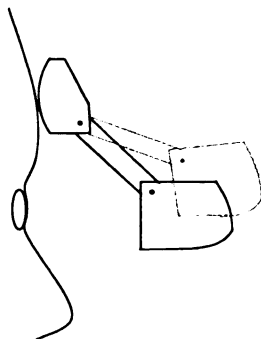


Figure 3.4: Possible Movement of the Sony Glasstron

is a bar connecting this contact point to the other display components. This allows for substantial movement of the optical components, both changing the location and pose.

### 3.4 Measurements used for Model

The most complicated component of the Glasstron is the curved mirror located at the front of the glasses. The values currently being used are from direct physical measurements.

The HMD is modeled with two spheres, two planes for the flat mirror, as well as a plane for the LCD screen. All the locations given are relative to the rear, the side facing the user, of the HMD. This allows for modeling movement of the HMD by moving this reference point. The standard configuration of the system has this reference point set at  $(20mm, 0mm, 0mm)$  in the world coordinate system.

| Surface         | Relative Location | Radius  |
|-----------------|-------------------|---------|
| Front of Lens   | -27.26mm          | 57.47mm |
| Rear of Lens    | -27.26mm          | 55.44mm |
| Front of Mirror | 19.24mm           | n.a.    |
| Rear of Mirror  | 17.74mm           | n.a.    |
| LCD Screen      | 0mm               | n.a.    |

Table 3.3: Values for Head Mounted Display

### 3.5 Considerations

This model is based on average information of the population. There are other factors that can influence what happens to rays that enter the eye. A user's physiology will necessarily vary from the standard. Also, many people wear corrective lenses to allow for normal vision. Lastly, the eye has the capability to change shape to focus on objects at different depths. All of these factors are currently not part of the computational model that is being used [8, 46]. They can be included in the model in the future. The goal of this thesis is to determine what is possible assuming a given eye and display combination. The variances from different users will necessary need to be accommodated in a calibration process.

### 3.6 Components as a System

All of these optical components are combined to form a system which is responsible for presenting information to the user. In order to simulate different usage situations, modifications can be applied to the system. These modifications can take the form of changing the location of the HMD or rotating the eye. All of this information is then used by the methods presented in Chapter 4. The combination of the two provides a means to answer questions about the entire system. The HMD and eye are no longer viewed as individual objects, but the combination as one image formation system.

## CHAPTER 4

### METHODS

The primary goal of this thesis is a derivation of a model for the eye and optical see-through head-mounted display as a system. Such a model will allow us to make several critical determinations. How close is the common pin-hole camera model to the actual model required by the system? How much error does the pin-hole camera model introduce? How much error is introduced when the eye moves if a static projection model is utilized. This section describes the mathematical methods that have been used to derive a model for the eye-display system, determine an optimum projection model, and determine the errors that will be introduced in common usage.

#### 4.1 Overview of Methods

In order to answer questions about the system, all the components must be modeled mathematically. These models provide a means of determining the behavior of light as it travels into the eye. The light refracts with each surface in the system as it enters the eye. Next, an initial ray is created that will refract with the surfaces of the system, producing a location on the iris. This initial ray is adjusted using a search technique to find a ray that will pass through the center of the iris.

This ray is the first step in creating the mappings that are the final result of the methods in this chapter. A mapping from the real world to the display is created by reversing this ray so it is traveling out of the eye. It is then reflected off the curved mirror of the HMD and finally intersected with the display, yielding a location on the display. To create a map from the display to the real world a different initial ray is

need. It is computed by adjusting the order in which the components are dealt with. Again the result ray is reversed, and traced out into the real world.

## 4.2 Mathematical Models

There are two steps necessary to determine what will happen to a ray of light as it travels into the eye through different optical objects. The first question is: Where will the given ray strike a surface? After this is known, a new direction is produced as the light travels from one medium to another. This new direction must be determined as well. In order to determine the behavior of light entering the eye a technique known as ray tracing is used [11, 16]. In this system, mathematical equations are used to represent objects and rays are constructed. These rays are then traced into the scene to see what they will intersect with. Rays are represented by vectors and surfaces in the system are represented with spheres and planes.

Rays in the system are modeled as vectors using a point and direction representation. A simple representation is:

$$P = E + tD \quad (4.1)$$

where  $P$  represents any point on the ray,  $E$  is some starting point for the ray,  $D$  is the direction, and  $t$  is some distance in that direction. Typically,  $D$  is a normalized vector and  $t$  represents unit steps in the direction of the vector. This equation can also be expanded out to:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} + t \begin{bmatrix} x_d \\ y_d \\ z_d \end{bmatrix} \quad (4.2)$$

#### 4.2.1 Sphere Intersection

The eye model presented in section 3.2 as well as the main mirror of the HMD use a spherical shape for all the surfaces. This can be modeled mathematically using a simple equation of a sphere with center at  $(x_c, y_c, z_c)$  and radius  $r$ .

$$(x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2 - r^2 = 0 \quad (4.3)$$

To find the intersection of a ray with a sphere, simply substitute the values from equation 4.2 into equation 4.3. This yields a quadratic equation that can be solved using the quadratic formula. If the discriminant is less than zero, no solution is possible. This means the two objects do not intersect. If the discriminant is greater than zero, there will be two solutions. One being the location where the ray enters the sphere and one when it leaves. Depending on the surface being modeled the correct solution can be chosen. If the surface is convex the closer point should be chosen, otherwise the further one is used.

#### 4.2.2 Plane Intersection

The other possible object a ray may intersect with is a plane. A plane is modeled using a point on the plane  $Q$  as well as a normal  $N$ . A point  $P$  is on a plane if  $N \cdot (P - Q) = 0$ . Substituting equation 4.1 in for  $P$  above yields the following.

$$t = \frac{N \cdot (Q - E)}{N \cdot D} \quad (4.4)$$

If  $t \geq 0$  then the ray will intersect the plane at  $E + tD$ .

### 4.2.3 Direction Computation

Now that a location of intersection is known, the next step is to calculate a new direction vector based on the angle of incident and the index of refraction of the two materials involved. This is done using vector methods. In the following figures the surface is shown to be curved, but the same methods work for both curved and planar surfaces. The only difference is the planar surfaces have the same normal at all intersection points, and a sphere normal is different at each point on its surface. The direction of the ray entering the surface is known. In Figure 4.1 this is labeled

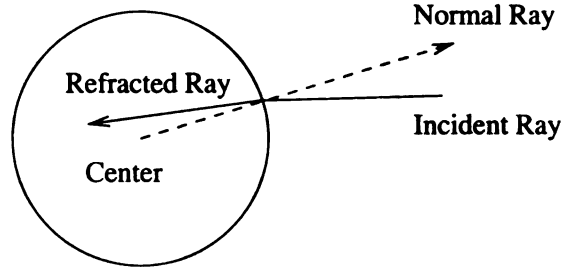


Figure 4.1: Refraction

as *Incident Ray*. Also known from the computation in section 4.2.1 or 4.2.2 is the location the ray intersects a surface. If the surface is a sphere a normal can then be created by forming a vector from the center of the sphere to the intersection point. If the surface is planar, the normal has already been given. This is labeled *Normal Ray*. In order to determine the direction of the *Refracted Ray*, one must use Snell's Law [21]. Snell's Law applies to light traveling from one medium into another. It is stated as

$$n_1 * \sin(\alpha) = n_2 * \sin(\beta) \quad (4.5)$$

where  $\alpha$  is the incident angle,  $\beta$  the refracted angle,  $n_1$  the original index, and  $n_2$  the index of the second medium. These angles are measured in reference to the computed normal of the surface.

The next step is calculating the incident angle to use in equation 4.5. The dot

product can be used to determine the incident angle. The dot product, defined in equation 4.6, has the property that it is equal to the product of the two vectors and the cosine of the angle between them, shown in equation 4.7 [20].

$$a \cdot b = a_x * b_x + a_y * b_y + a_z * b_z \quad (4.6)$$

$$a \cdot b = |a||b| \cos(\alpha) \quad (4.7)$$

Using the dot product,  $\alpha$  from equation 4.5 can be computed. From here the value of  $\beta$  can be determined. The value of  $\beta$  is the angle between the refracted ray and the surface normal. Even though the angle with respect to the normal is determined, the final refracted direction is still not known. This final direction must lie in the same plane as both the normal and original direction. This is done by first constructing a right triangle using the incident ray and the normal. The dot product can be used again to construct the right triangle. In a right triangle the cosine is the ratio of the adjacent side and hypotenuse. If the lengths of both the *dir* and *n* vectors is set to 1, the dot product is exactly the cosine of the angle. The length of *dir* from Figure 4.2 is set to be 1, and *n* is set to the value of the dot product. These measurements produce a right triangle. The third side of the triangle in Figure 4.2 is *anglevec* and

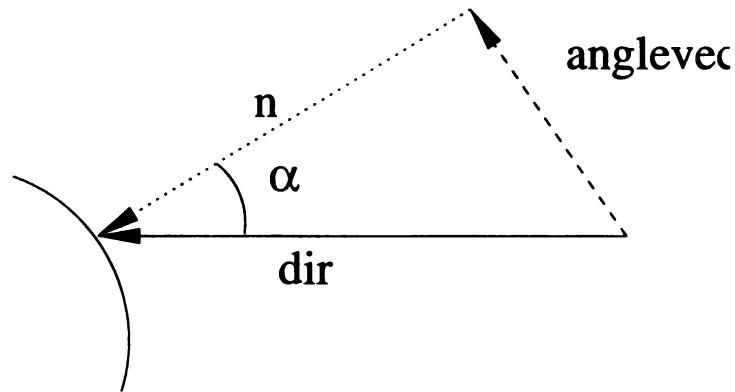


Figure 4.2: Incident Angle

is computed as  $dir - n$ . A second right triangle is constructed using *n* and a scaled

version of *anglevec* to create the final resultant vector with the correct value of  $\beta$ . The

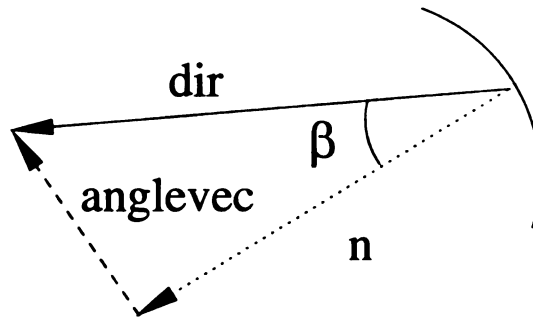


Figure 4.3: Resultant Angle

value of  $\beta$  in Figure 4.3 is known and thus the length of *anglevec* is scaled to  $\tan(\beta)$ . This produces a right triangle with the correct angle measurement. Therefore the final direction is  $n + \text{anglevec}$ .

#### 4.2.4 Subsequent Surfaces

The calculations in sections 4.2.1, 4.2.2, and 4.2.3 enable the computation of the result of a ray passing through one refractive surface. There are many surfaces involved in the Display Eye system. Depending on the desired behavior being modeled in the system, the different surfaces may be involved at different times. Therefore the computations have been presented in a way that allows for changes in order. The computation for a single surface takes a ray as input and produces a resultant ray as output. This new ray is the combination of the intersection point and the refracted direction. The output can be used as input for the next surface in the model. The different behaviors of the system and the necessary order of computation will be discussed later.

#### 4.2.5 Pin-Hole Camera Model

OpenGL, DirectX, and many other graphics systems use a pin-hole camera model to create the graphics that are displayed to the user. It is this rendering that is

critical in determining the error contributed by the display eye system when these systems are utilized unmodified. The model is based on the idea of an infinitely small hole that allows only one ray of light from any location through its opening. This one ray can then be used to determine where the light from a location will show up on the image plane. This concept is shown in Figure 4.4 in a simplified 2 dimensional

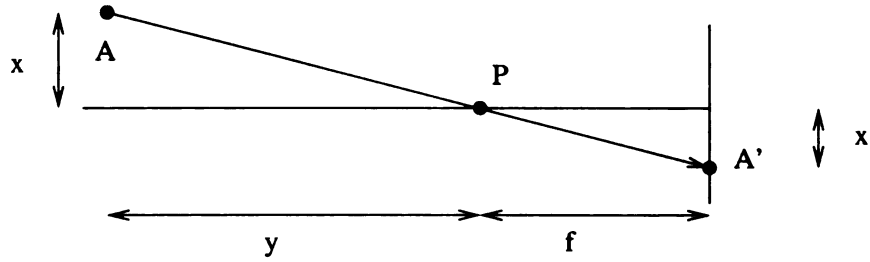


Figure 4.4: 2 Dimensional Pin Hole Camera

representation. The pin hole is labeled  $P$ . A ray of light from point  $A$  travels through  $P$  and strikes what is known as the image plane at  $A'$ . This image plane is a distance of  $f$  away from the pin hole, a measure called the focal length. The location of  $A$  is known to be  $(x, y)$  and  $A'$  is  $(x', f)$ . These measures are directly related to as follows,  $\frac{x}{y} = \frac{x'}{f}$ . Since the only unknown is  $x'$  this is simplified to  $x' = \frac{xf}{y}$ . The same method works for generalization to 3 dimensions by simply repeating the process for the  $z$  direction. By this method a location on the image plane can be determined for any point in space [1]. The eye is significantly more complex than this simple model and has been an open question as to how well the pin hole camera model approximates the relationship between locations in space and locations in the projection window for the HMD.

OpenGL must still do many more computations in order to produce a final image. The calculations discussed above assumes the pin hole of the camera is located at the origin and the image plane has the  $y$  axis as its normal. This is not the case in practice, since the camera may be anywhere in space, and the direction of the camera can change. The points that make up a scene must be translated according to the

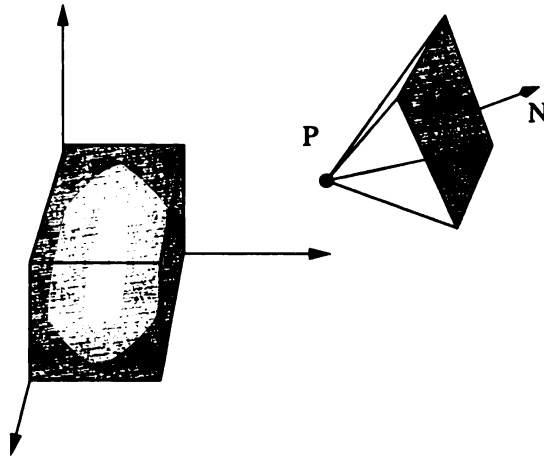


Figure 4.5: 3 Dimensional Camera

location and pose of the camera. This means means the pin hole of the camera is translated to the origin as well as orienting the normal on the  $y$  axis. Once this is done, the translated point may be rendered accordingly.

### 4.3 Modeling Behavior of the Entire System

There are two major questions that need to be answered by the model of the Display Eye system. These two questions are the central theme of this thesis. First, given a location in 3D space, what 2D point in the display does it correspond to (the project operation)? Second, given a point in the 2D display, what 3D points in space does it correspond to (the unproject operation)? These questions obviously enjoy an inverse relationship. In order to answer these questions a more basic question must first be solved. What happens to a ray of light from a point in space as it passes through the components of the system? This question is easily answered by using the original ray and refracting it with the first surface of the system. This will yield a second ray to be refracted with the next surface. This can be repeated for all the components of the system in order, until finally the ray intersects with the retina of the eye.

### 4.3.1 Mapping Real World to Display

The input to the real world to display mapping is a point in space. An initial ray is created and traced into the eye. The ray is created by directing a ray at the center of the iris from the initial point. The surfaces involved, in order, are: the front of the curved mirror, the rear, the front of the flat mirror, the rear, the front of the cornea, the rear of the cornea, and finally the iris. The method in Section 4.4 provides a way to adjust this ray so it passes through the center of the iris. This ray is then reversed and reflected off the curved mirror of the HMD. When the ray is intersected with the sphere modeling the mirrored surface, it forms an incident angle. The angle of incident is equal to the angle of reflection. Therefore it is possible to compute the angle of reflection, given the angle of incident. In Figure 4.6 the vector labeled *inc* is

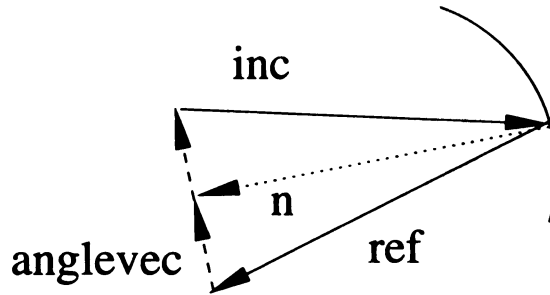


Figure 4.6: Reflection in a Curved Mirror

the incident direction and is known. Using methods from Section 4.2.3, the *anglevec* is computed. Thus the reflected vector must be  $-1 * (ref + 2 * anglevec)$ . This forms a new ray when combined with the original point on the curved mirror. This ray is finally intersected with the plane of the LCD to produce a point on its surface. This point is the result of the mapping of a real world point into the HMD.

### 4.3.2 Mapping Display to Real World

The model must also be able to generate 3D points in space given a 2D location in the HMD. This is not as specific as the first function. In mapping onto the LCD

surface, there was a final surface that could be intersected with. This is not the case when trying to find a location in space. Thus the result of this mapping is a vector directed away from the HMD. This vector can be intersected with any plane in space to yield a point result.

The method for creating this mapping follows the same steps as the previous mapping. The input is a point on the LCD plane. First a starting ray must be created that will be used as input for the selection process in section 4.4. This ray is created by first defining a direction that begins at the center of the iris and ends at the input point on the virtual display. The initial ray is defined as having a starting at the input point and the direction previously defined. This ray is used because it will intersect will the appropriate surfaces. The selection process needs a method of

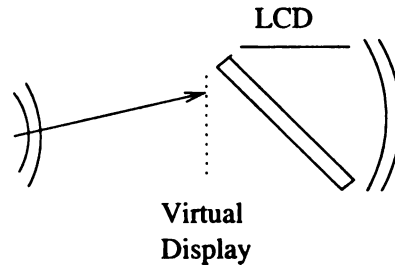


Figure 4.7: Initial Ray Selection for HMD to Real World Mapping

intersecting a ray with the iris, in the form of a list of interactions between the ray and the optical components. In this case the ray must first be reflected off the curved mirror of the HMD. This is done using the same method as finding the reflected ray from the previous section. The reflected ray is then refracted with the two surfaces of the flat mirror and the first two surfaces of the eye. It is then intersected with the iris. Once the appropriate ray has been selected, it can be reversed and refracted well all the surfaces in reverse order. The final result being the output vector.

## 4.4 Ray Selection

An object in space will theoretically reflect the light that hits it in an infinite number of directions. There are any number of rays that contribute to forming images in the eye. Therefore a single ray must be selected that is representative of a path into the eye. This is done by determining a ray that will go through the center of the iris. This process provides a method of computing an intersection point on the iris plane given a ray, as well as an initial ray. The method of computing this intersection is such that a closed form solution is not feasible. Therefore no inverse could be computed that would yield a ray from the center of the iris that would intersect with a given point in space. Therefore another method for finding a ray from a given point through the center of the iris is needed.

### 4.4.1 *Searching*

A search method is needed to find an appropriate ray. Typically search algorithms are designed to find an object in one dimension. For example searching for the largest value in a list or checking to see if an element is in an ordered list. The search space involved in finding a desired ray is two dimensional, since a rays direction may be changed horizontally and vertically. In order to find the desired ray a differential technique is used. It is referred to as differential since it uses a local rate of change to compute a new value for the next iteration. First an  $\epsilon$  value is chosen as an acceptable amount of error. The process is started by using the provided initial ray. It is not terribly important where this ray may end up, only that it strikes the optical surfaces involved. This ray is intersected with the components until it passes through the iris. In Figure 5.1 this point on the plane of the iris is labeled *A*. Then two new rays are formed, one by tilting the starting ray up slightly and another by tilting it to the side. These new rays are traced into the iris and are marked by  $z'$  and  $y'$ . They will both

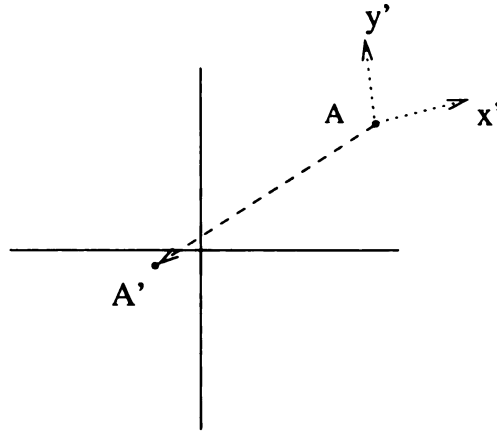


Figure 4.8: Search Vectors

result in some change in location from the original intersection point. Some linear combination of the two changes are added to  $A$  so the result is  $(0,0)$ , the center of the iris. The same linear combination of the changes made to produce the new rays is added to the original ray. This produces an approximation of a ray that will pass through the center of the iris. This new ray is then traced into the eye as the previous ray and its intersection is marked as  $A'$ . The error is calculated as the distance from the origin. The process is repeated with this new ray until the error is less than the given  $\epsilon$ .

## **CHAPTER 5**

### **RESULTS**

The models and methods presented in Chapters 3 and 4 provide the necessary information and techniques such that given a point in space a ray can be constructed that will pass through the center of the iris after being refracted by the optical components in front of the iris. This information in itself is not useful since it tells nothing about error. Tests needed to be designed that would use this capability to provide data that it is relevant. These tests model real world situations that a user of an AR system would confront.

#### **5.1 Graphical Rendering Error**

The first research question is how much error is due purely to the simplified graphics rendering of the pinhole model and the introduction of optical components in front of the eye. In order to answer this question an optimum situation is designed. This situation makes three assumptions in order to isolate error to the display eye system. The first is that the data from the tracking system is going to be completely free of error. This allows us to specify the exact position of the eye and assume that perfect tracking would provide this information accurately. The second is assumption of an optimal calibration. The calibration process estimates the location and focal length for a camera that represents the eye. This estimate is needed since highly precise measurements of the location and angle of the display with respect to the eye are not available. The third assumption is that the location and direction of the eye is predefined. The eye is set to be located directly in line with the center of the virtual display and a radius of the curved mirror of the HMD.

### 5.1.1 Experiment Layout

The first step in laying out all the experiments is defining a world coordinate system. This is needed to allow relative measurements of the eye, HMD, and real world points to be defined in meaningful ways compared to the others. The chosen coordinate system places the origin at the center of the ocular globe. The ocular globe is the main component of the eye that rotates inside the eye socket of the skull. This object is modeled by surface 5 from Table 3.2. The second component of a coordinate system that is needed is an orientation of the axis. The X-axis is placed pointing directly out of the eye and is normal to the other refractive surfaces of the eye. The optical

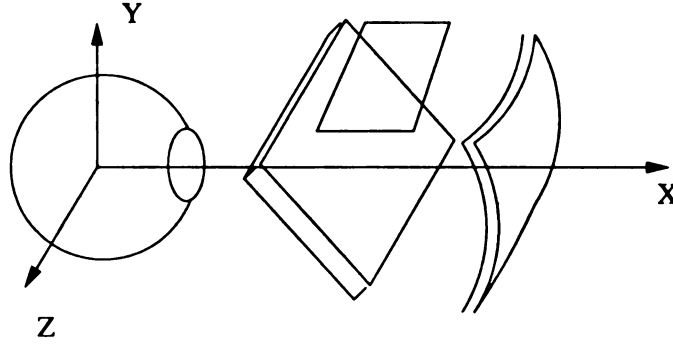


Figure 5.1: Layout of Optical Components for Extra-Fovea Points

components of the HMD are placed in such an orientation that the X-axis is normal to the curved mirror and passes through the center of the virtual display.

The next step is determining the optimal camera calibration that will be used to compare points rendered by the system to their optimal location. This calibration is created by defining a correct rendering in a small scale situation. In order to make calculations easier the image plane of the pin-hole camera is moved in front of the origin which represents the pin hole. The image plane is represented by a plane which is orthogonal to the X axis. Rays are formed that connect the points in the real world to the origin. These rays intersect the image plane at some point. The display eye model is used to generate these first points on the image plane. Then the properties

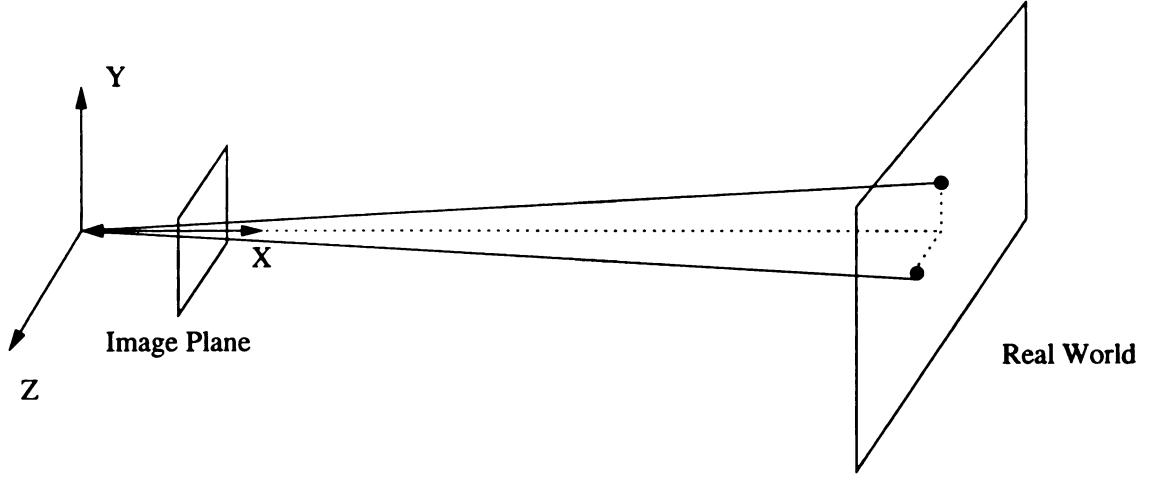


Figure 5.2: Points used to create optimal configuration

of a pin hole camera can be determined that would produce the same locations on the image plane. This camera can then be used on other points and the results compared to those from the display eye model.

The smallest increment that the display is capable of representing is a pixel, therefore it is chosen as the small range from which the pin hole camera is defined. The LCD in the display is  $12.7mm$  wide and has a horizontal resolution of 800 pixels. This leads to the fact that each pixel is  $0.015875mm$  wide. Computations using the display eye model show that at a range of  $1200mm$ , the range of the display, a point at  $(0.6956, 0)$  in the real world, will map to  $(0.015875, 0)$ , a pixels width, in the image plane. Using the equation for the pin-hole camera, this leads to the following equation.

$$\frac{0.6956}{1200} = \frac{0.015875}{f} \quad (5.1)$$

Therefore the focal length to be used in the optimal camera is  $27.386mm$ . The only other concern is an offset in the Y direction. The flat mirror of the HMD results in a shift of the rays passing through it. The point  $(0, 0)$  in the real world maps to  $(0, 0.01613)$  in the display. This can be accommodated by shifting the camera down  $0.01613mm$ . These two measurements, along with the normal defined as the X axis,

completely describe the pin hole camera. Points in space can be rendered to the image plane using the following equations.

$$z_{ip} = \frac{z * f}{x} \quad (5.2)$$

$$y_{ip} = \frac{y * f}{x} + 0.0124555 \quad (5.3)$$

This thesis is comparing two methods of determining an appropriate LCD display pixel from a point in space. The first is through the use of optical analysis using the methods discussed in Chapter 4. This method will produce a location in the LCD that will need to be illuminated in order to overlay a real world point. The second method is using the pin hole camera approximation model with the parameters determined above. These two methods both produce a location in the LCD for a point in space. Measurements of distances on the LCD are not meaningful in themselves. To say there is an error of  $0.1mm$  in the display is not as meaningful as an error of  $1cm$  in the real world. For this reason one needs to find the location in space that a point in the display will be perceived. This is done using the method from Section 4.3.2. The point rendered using the pinhole camera can be mapped back out to a real world location. The final result is a pair of points, the original and the perceived location of a pinhole camera rendering. When compared, the difference is the error due to the properties of the display eye system. The results of applying both models to a range of points in space are shown in Figure 5.3 and Table 5.1. The following locations are all at distance of  $1200mm$  in front of the eye. The  $X$  and  $Y$  ranges were chosen by tracing the corners of the LCD out into the real world.

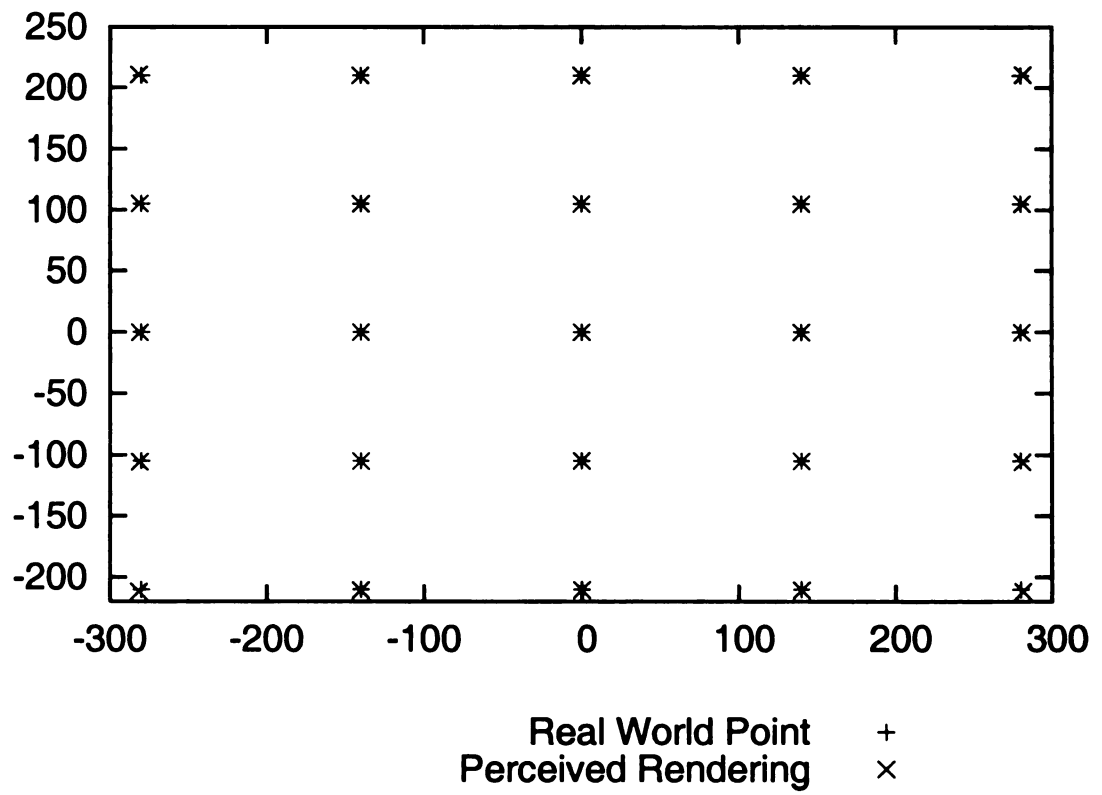


Figure 5.3: Comparison of Real World Points and Perceived Rendering

|           |          |                |                |
|-----------|----------|----------------|----------------|
| Max Error | 2.292    | Mean Error     | 0.396          |
| $X_{rw}$  | $Y_{rw}$ | $X_{rendered}$ | $Y_{rendered}$ |
| -280      | -210     | -281.567       | -211.673       |
| -140      | -210     | -140.329       | -210.697       |
| 0         | -210     | 0              | -210.372       |
| 140       | -210     | 140.329        | -210.697       |
| 280       | -210     | 281.567        | -211.673       |
| -280      | -105     | -281.061       | -105.748       |
| -140      | -105     | -140.078       | -105.143       |
| 0         | -105     | 0              | -104.942       |
| 140       | -105     | 140.078        | -105.143       |
| 280       | -105     | 281.061        | -105.748       |
| -280      | 0        | -280.897       | -0.320978      |
| -140      | 0        | -139.997       | -0.0790623     |
| 0         | 0        | 0              | -3.75409e-10   |
| 140       | 0        | 139.997        | -0.0790623     |
| 280       | 0        | 280.897        | -0.320978      |
| -280      | 105      | -281.068       | 105.018        |
| -140      | 105      | -140.086       | 104.905        |
| 0         | 105      | 0              | 104.866        |
| 140       | 105      | 140.086        | 104.905        |
| 280       | 105      | 281.068        | 105.018        |
| -280      | 210      | -281.57        | 210.664        |
| -140      | 210      | -140.341       | 210.206        |
| 0         | 210      | 0              | 210.051        |
| 140       | 210      | 140.341        | 210.206        |
| 280       | 210      | 281.57         | 210.664        |

Table 5.1: Values of Real World Point and Perceived Rendering

## 5.2 Isolated Eye Movement

The retina of human eye has a small portion in which the ability to sense light is much greater than the rest of the retina. This region of the retina is known as the fovea and has a range of approximately  $5^\circ$  about the center of the eye[7]. This results in humans having a small region of clear vision and a loss of detail on the periphery of their vision.

The fovea is a small portion of the retina but provides the brain with the most detailed information. Therefore to sense the environment in detail, the eye is moved

so that light from different portions of the world fall on the fovea. There are two ways this can be accomplished, either by moving the head or by moving the eye itself. Since the location and pose of the head is tracked this movement is input into the AR system and adjustments are made accordingly. The movement of the eye is typically not tracked and it has been a question of some debate as to how much affect moving the eye has on the calibration. If no eye tracking is used, the image presented to the user remains the same when the eye is moved. Thus it is possible that the movement of the eye in its socket produces some amount of error.

All of the optical components that are involved when the eye moves are modeled in the system. Therefore the situation examined in Section 5.1 can be generalized to allow the eye to move in its socket. The movements of the eye are rotations about a fixed point. The fixed point is the center of the sphere that models the retina [7]. When this movement takes place, the tracking system does not register any change. Therefore the image presented to the user will be the same as before the eye was moved. It is this fact that allows for the introduction of error. The magnitude of the error is determined by the methods laid out above. The only new technique that is needed is a method of rotating the eye. This is done using a linear transformation. A matrix is defined that will accomplish the desired rotation. This matrix is applied to the different components of the eye. For the spheres, this is done by multiplying the transformation and the center. The iris is modeled using a plane, thus the transformation must be multiplied with both the point on the plane and the normal.

The configuration of the system changes slightly for this situation. The HMD has not moved so the same pixels of the LCD are lit. The difference is only the location that these pixels may be perceived. Therefore the method from Section 4.3.2 can be used on these display locations. The only change that is made is rotating the components of the eye. The actual display area of the HMD occupies approximately

26° of the field of view, therefore a rotation of 13° from the center would be the maximum angle that would allow light from the display to strike the fovea. This degree of rotation is also assumed provide the largest amount of error, therefore this angle is used to demonstrate the effect of rotating the eye. The results from rotating 13° about the *Y* axis are shown in Figure 5.4 and Table 5.2. Rotation about the *Z* axis, looking up with the eye) show a similar magnitude of error.

|           |          |                |                |
|-----------|----------|----------------|----------------|
| Max Error | 2.382    | Mean Error     | 1.353          |
| $X_{rw}$  | $Y_{rw}$ | $X_{rendered}$ | $Y_{rendered}$ |
| -280      | -210     | -280.224       | -210.787       |
| -140      | -210     | -138.835       | -209.907       |
| 0         | -210     | 1.2211         | -209.897       |
| 140       | -210     | 140.763        | -209.933       |
| 280       | -210     | 280.588        | -210.824       |
| -280      | -105     | -279.379       | -105.343       |
| -140      | -105     | -138.189       | -104.788       |
| 0         | -105     | 1.67094        | -104.746       |
| 140       | -105     | 141.015        | -104.805       |
| 280       | -105     | 280.643        | -105.37        |
| -280      | 0        | -279.392       | -0.311631      |
| -140      | 0        | -138.12        | -0.073537      |
| 0         | 0        | 1.82124        | 0.00211766     |
| 140       | 0        | 141.245        | -0.0780219     |
| 280       | 0        | 280.952        | -0.320293      |
| -280      | 105      | -279.39        | 104.637        |
| -140      | 105      | -138.199       | 104.565        |
| 0         | 105      | 1.67057        | 104.676        |
| 140       | 105      | 141.024        | 104.573        |
| 280       | 105      | 280.652        | 104.646        |
| -280      | 210      | -280.232       | 209.806        |
| -140      | 210      | -138.849       | 209.434        |
| 0         | 210      | 1.22022        | 209.585        |
| 140       | 210      | 140.776        | 209.45         |
| 280       | 210      | 280.594        | 209.825        |

Table 5.2: Values for Rotation of 13°

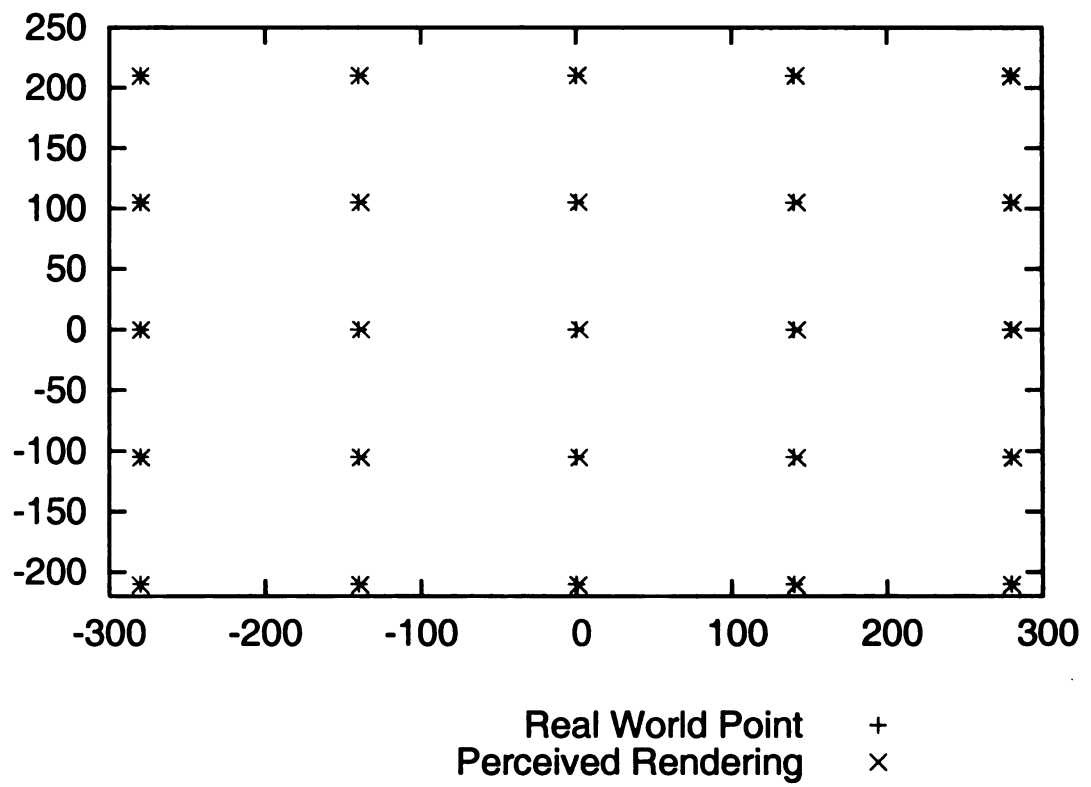


Figure 5.4: Graph of Error Based on a Rotation of  $13^\circ$

### 5.3 Movement of the Head Mounted Display Relative to the Eye

When a user begins using the AR system, they must first calibrate the system. Part of this process is determining the relative location of the eye to the HMD. After this is set in calibration, it no longer changes. It is, however, possible to move the HMD relative to the eye while using the system. This would be unnoticed by the tracking software and thus not result in a change in the image presented to the user. Similar to rotating the eye, this could also be a source of error. Any possible error introduced by movement of the display relative to the eye is of great interest in repeatability of calibrations. How accurately would a display need to be replaced on the head if a calibration is to be reused?

In order to model this situation, the components of the HMD must be moved. This is done by simply adding the offset to each location coordinate for the HMD components. The normal of the half silvered mirror remains the same. The lit pixels of the HMD change location relative to the eye, thus the offset must be added to the pixel location before it is sent to the process discussed in Section 4.3.2. The result of this computation is a perceived location in space.

The viewing window of the HMD is approximately  $30mm$  wide, therefore a movement on the scale of  $10mm$  would nearly move the eye out of this frame. Thus a change on a smaller scale is used to demonstrate the effects of moving the display. The values in Figure 5.5 and Table 5.3 are the result of moving the display up  $3mm$  in the  $Y$  direction and  $2mm$  closer to the eye. These measurements were chosen to simulate the movement of the display along the nose of the user.

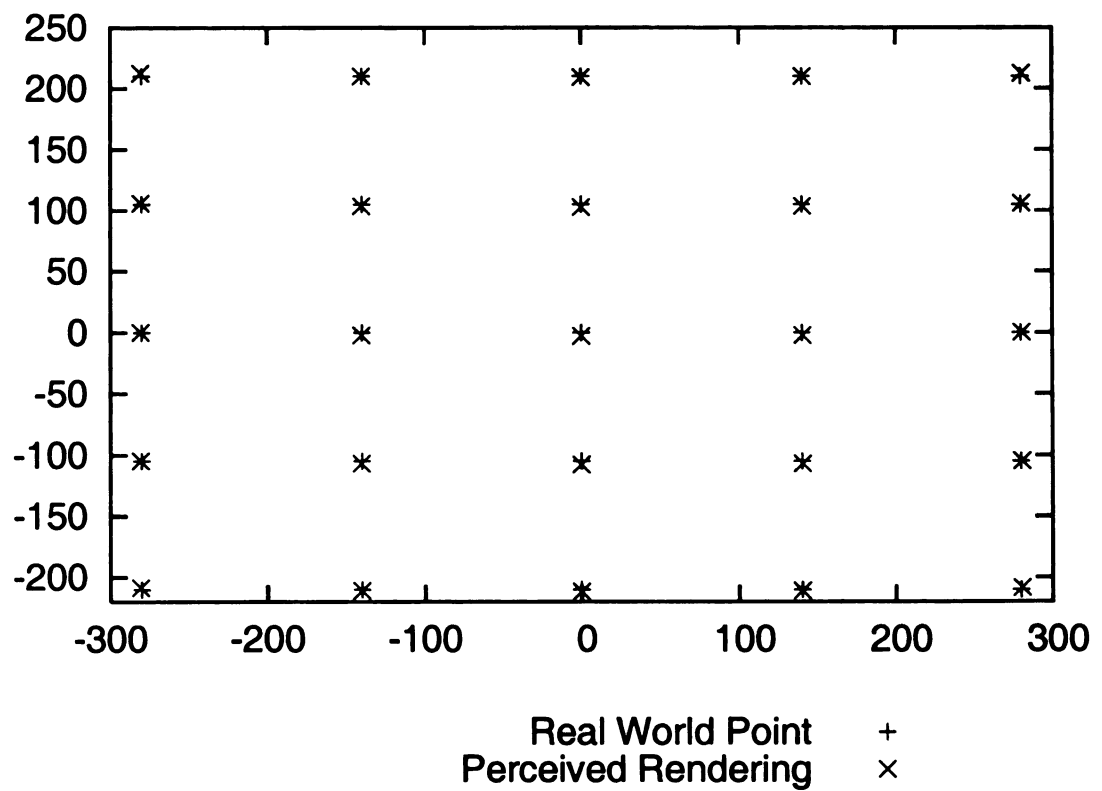


Figure 5.5: Graph for Moving the Display

|           |          |                |                |
|-----------|----------|----------------|----------------|
| Max Error | 11.03    | Mean Error     | 2.092          |
| $X_{rw}$  | $Y_{rw}$ | $X_{rendered}$ | $Y_{rendered}$ |
| -280      | -210     | -281.55        | -209.157       |
| -140      | -210     | -140.321       | -210.741       |
| 0         | -210     | 0              | -211.276       |
| 140       | -210     | 140.321        | -210.741       |
| 280       | -210     | 281.55         | -209.157       |
| -280      | -105     | -281.025       | -104.715       |
| -140      | -105     | -140.059       | -106.689       |
| 0         | -105     | 0              | -107.355       |
| 140       | -105     | 140.059        | -106.689       |
| 280       | -105     | 281.025        | -104.715       |
| -280      | 0        | -280.842       | 0.169166       |
| -140      | 0        | -139.969       | -2.17864       |
| 0         | 0        | 0              | -2.9699        |
| 140       | 0        | 139.969        | -2.17864       |
| 280       | 0        | 280.842        | 0.169166       |
| -280      | 105      | -280.998       | 105.918        |
| -140      | 105      | -140.049       | 103.216        |
| 0         | 105      | 0              | 102.306        |
| 140       | 105      | 140.049        | 103.216        |
| 280       | 105      | 280.998        | 105.918        |
| -280      | 210      | -281.487       | 212.926        |
| -140      | 210      | -140.297       | 209.891        |
| 0         | 210      | 0              | 208.87         |
| 140       | 210      | 140.297        | 209.891        |
| 280       | 210      | 281.487        | 212.926        |

Table 5.3: Values for Moving the Display

## CHAPTER 6

### DISCUSSION AND CONCLUSIONS

The values presented in Chapter 5 deal with specific usage situations. This chapter will discuss what these results mean in an application setting as well as their relation to observed characteristics of the HMD. Also, it will present some possible methods to reduce error in optical see-through display applications.

There are three situations that are discussed. In order to validate the results of the system modeling, real world measurements were taken. These situations were reproduced in the laboratory using the HMD and a calibrated camera. The information from this calibration is used to determine what pixel differences in the image are equivalent to in the real world.

#### 6.1 Pin Cushion Effect

The first result is the pin cushion effect that the display has on the images it presents. Points that are near the edge of the LCD face a certain amount of distortion, which increases the further they are from the center. This causes straight lines in the display to appear curved when compared to the real world. This matches what is seen through the HMD. The image in Figure 6.1 was taken by a camera looking through the HMD. The corner points are connected with a line. Then the distance from the middle point to this line is measured. The center top of the display is measured to be one pixel below this line which works out to be  $1.68mm$  at  $1200mm$ . This closely corresponds to the values from the models. The data in Table 5.1 shows a difference of  $0.613mm$  between the height of the top center and corner points. Since the measured difference is so small, a difference of one pixel would change the measurement significantly. The

images taken through the HMD are somewhat blurry, thus such an error is possible. A higher resolution camera could provide more detail but it may not be practical to position the HMD in front of it.

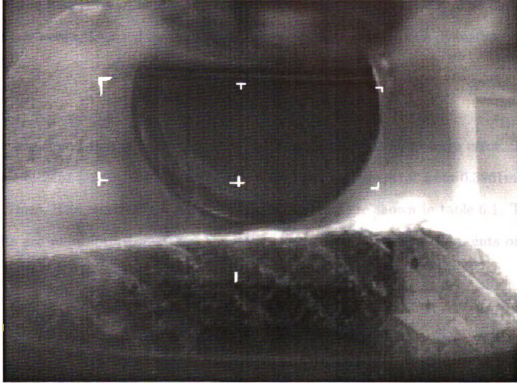


Figure 6.1: Real World Pin Cushion Example

At the corners of the display, which show the greatest amount of error, the magnitude is  $1.7mm$  at a range of  $1200mm$ . In the experiments conducted by McGarrity [26], measurements were taken in a working space no larger than arms length, or about  $700mm$ . At this range the error scales to  $0.99mm$ . The results from McGarrity showed minimum errors on the order of  $10mm$ , thus small improvement could be made by adjusting for this rendering error. It is clear that the majority of the error in the McGarrity study was due to the design of the calibration method, a human-computer interaction problem that needs a great deal of further study.

The error is larger the further a point is away from the center of the display, this error is modeled as radial distortion. It is possible to warp an image to compensate for

radial distortion. If this warp were applied to the input to the HMD before it is sent, the radial distortion could be canceled out. This could be accomplished with a more complex rendering method [19, 37, 44]. A simple radial warp was designed that would cancel the distortion. The distortion value is defined in equation 6.1. The distorted locations were calculated by multiplying the original values by this distortion value.

$$1 + \kappa * \sqrt{|x|} + \sqrt{|y|} \quad (6.1)$$

It was found that a value of  $\kappa = -0.000834$  minimized the average error. For a sample of 2451 points, this distortion decreased the average error from  $0.3961mm$  to  $0.3327mm$ . The results for a limited number of points are shown in table 6.1. These results are more impressive than the average error, showing how the points on the edge of the display are corrected.

## 6.2 Handling Eye Movement

The results in the previous section are concerned with points that are a large distance away from the center of the display. It also assumes the user does not move their eye to examine these points. The more likely case is that the user will rotate their eye in order to focus this area on their fovea.

The most extreme case of this movement was modeled in Section 5.2. The eye was moved to focus on the point  $(280, 0)$ . Previous to this movement, the rendering would be perceived at  $(280.897, -0.321)$ , and afterward error is reduced. Now the rendering is perceived at  $(280.652, -0.320)$ . The change in perception after the rotation is less than  $0.5mm$ , a small measurement at a range of  $1200mm$ . For this reason, the orientation of the eye does not contribute significantly to the error of the system. These calculations correspond to the measured changes from simulating rotation of the eye. The image is shown in figure 6.2. The center mark of this image is at

| $X_{rw}$ | $Y_{rw}$ | $X_{distorted}$ | $Y_{distorted}$ |
|----------|----------|-----------------|-----------------|
| -280     | -210     | -280.446        | -210.826        |
| -140     | -210     | -139.861        | -209.992        |
| 0        | -210     | 0               | -209.985        |
| 140      | -210     | 139.861         | -209.992        |
| 280      | -210     | 280.446         | -210.826        |
| -280     | -105     | -280.097        | -105.382        |
| -140     | -105     | -139.687        | -104.848        |
| 0        | -105     | 0               | -104.806        |
| 140      | -105     | 139.687         | -104.848        |
| 280      | -105     | 280.097         | -105.382        |
| -280     | 0        | -280.274        | -0.318334       |
| -140     | 0        | -139.776        | -0.0779452      |
| 0        | 0        | 0               | 5.09483e-05     |
| 140      | 0        | 139.776         | -0.0779452      |
| 280      | 0        | 280.274         | -0.318334       |
| -280     | 105      | -280.107        | 104.662         |
| -140     | 105      | -139.697        | 104.615         |
| 0        | 105      | 0               | 104.732         |
| 140      | 105      | 139.697         | 104.615         |
| 280      | 105      | 280.107         | 104.662         |
| -280     | 210      | -280.452        | 209.831         |
| -140     | 210      | -139.874        | 209.508         |
| 0        | 210      | 0               | 209.668         |
| 140      | 210      | 139.874         | 209.508         |
| 280      | 210      | 280.452         | 209.831         |

Table 6.1: Values for Radial Distortion Correction

(292,228), while the center mark in figure 6.1 is (293,230). The most significant result is that the center changed 2 pixels in the  $Y$  direction. This could be a result of moving the setup to simulate the situation.

### 6.3 Relative Movement of the HMD

One might expect that small movements of the HMD with respect to the eye would be a large error contributor. Since the display is so close to the eye, it would seem that a small change would scale to a large change at a greater distance. However,

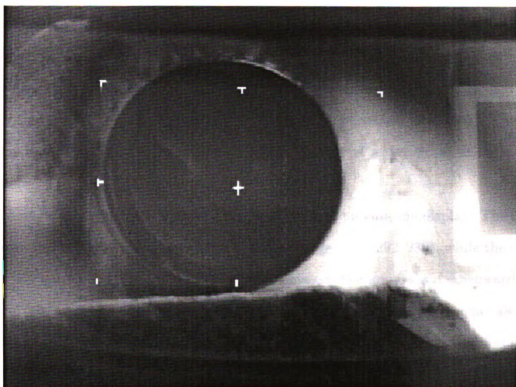


Figure 6.2: Simulated Rotation of the Eye

the optics of the HMD were created to mimic a large display some distance away from the user. Therefore when the display is moved small distances around the eye the result is closely correlated with moving a large display that same small distance. This phenomenon is only exhibited for a small range. As the changes become larger, the error introduced becomes much greater. For example a movement of  $10mm$  up in the  $Y$  increases the mean error to  $25.89mm$ , while a movement of  $3mm$  gives a mean error of  $1.63mm$ .

In the case of a movement in Section 5.3 the point  $(0, 0)$  was perceived at  $(0, -2.9699)$ . This change, though a change in the opposite direction of the movement, is not as drastic as what may have been expected. If this movement were scaled by the same factor that the LCD is apparently scaled, it would be perceived at  $(0, 180)$ . Thus the display is shown to closely approximate a large display at a significant distance from the user.

These calculations correspond to the changes from moving the display. The image is shown in Figure 6.3. The center mark of this image is at  $(292, 230)$ , while the center mark in Figure 6.1 is  $(292, 228)$ . The change between the images is a downward shift of 2 pixels after moving the display. This difference of 2 pixels is equivalent to  $-3.36mm$  at  $1200mm$ . The system modeling predicts the difference would have been  $-2.9699mm$ .

There is an additional way the HMD may move on a user. Changing position of the HMD is seen to not have a drastic effect on what the user perceives. This is not that case when the pose of the HMD is changed. A small change in the angle of HMD in reference to the eye results in a drastic change in perception. This follows from the design of the display. It is designed to be similar to a large display a distance from the user. Changing the angle of the HMD can be compared to moving the display that many degrees along an arc at the virtual distance.

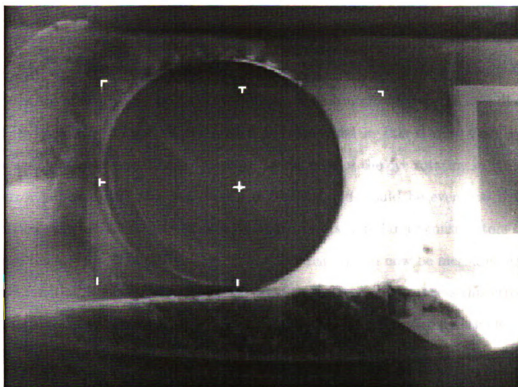


Figure 6.3: Simulated Display Movement

## 6.4 Conclusion

Prior to the work done in this thesis it was unknown how different usage situations of the AR system contributed to the overall errors that were measured. Every user would place the HMD on their head slightly differently. Some users would tightly secure the display and need to adjust as they continued usage. When running experiments, users could be asked to interact in some way with an object. They controlled where in the display this object was displayed by moving their head. It was not recorded where they chose to situate the object while interacting with it. Also it was noted that a pin cushion effect was noticeable, but was never measured. All of these unknowns fueled the question of what aspects of the AR system were contributing to the overall error.

There are three usage cases modeled and discussed in this thesis. Only two of these are significant contributors to error. In the case of the eye rotating in its socket, the error is not significant, less than a millimeter and would be even less if radial distortion were included. The other error contributors were larger contributors to the error of the system. The radial distortion of the display can now be measured and an inverse function found. If the inverse is added to the rendering process this error can be greatly reduced. The other significant contributor is the relative movement of the HMD. This error is difficult to track in real time. Precautions can be taken to limit the movement of the HMD, but during prolonged uses, it remains a likely occurrence. A HMD display designed to have a fixed angle would reduce the considerable error introduced by changes in pose of the display. An occasional recalibration done while using the system could correct for the error caused by movement of the display during usage [10, 12]. If these changes are implemented, the efforts of reducing error can then be focused on other components of the system.

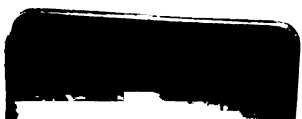
## BIBLIOGRAPHY

- [1] Edward Angel. *Interactive Computer Graphics: A Top-Down Approach with OpenGL*. Addison-Wesley, 2002.
- [2] R. Azuma. Making direct manipulation work in virtual reality. *SIGGRAPH Course Notes 30*, August 1997.
- [3] Ronald Azuma. Tracking requirements for augmented reality. *Communications of the ACM*, 36(7):50–51, 1993.
- [4] Ronald Azuma. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 4(6):355–385, Aug 1997.
- [5] Brian Barsky, Billy Chen, Alexander Berg Maxence Moutet, Daniel Garcia, and Staley Klien. Incorporating camera model, ocular model, and actual patient data for photo-realistic and vision realistic rendering. *Abstract in the Fifth International Conference on Mathematical Methods for Curves and Surfaces*, 2000.
- [6] Brian Barsky, Daniel Garcia, Stanley Klein, Woojin Yu, Billy Chen, and Sarang Dala. Rays (render as you see): Vision-realistic rendering using hartmann-shack wavefront aberrations. Internal Report, Mar 2001.
- [7] Hugh Davson. *Physiology of the Eye*. Little, Brown and Company, London, 1963.
- [8] Daniel Garcia. *CWhatUC : Software Tools for Predicting, Visualizing and Simulating Corneal Visual Acuity*. PhD thesis, University of California, Berkeley, Berkeley, California, 2000.
- [9] Y. Genc, F. Sauer, F. Wenzel, M. Tuceryan, and N. Navab. Optical see-through hmd calibration: A stereo method validated with a video see-through system. *Siemens Corporate Research*, 2000.
- [10] Michael Gleicher and Andrew Witkin. Through-the-lens camera control. *Computer Graphics*, 26(2):331–340, 1992.
- [11] Robert Goldstein and Roger Nagel. 3-d visual simulation. *Simulation*, 16(1):25–31, Jan 1971.
- [12] Li-wei He, Michael F. Cohen, and David H. Salesin. The virtual cinematographer: A paradigm for automatic real-time camera control and directing. *Computer Graphics*, 30(Annual Conference Series):217–224, 1996.
- [13] William Hendee and Peter Wells. *The Perception of Visual Information*. Springer, New York, 1997.

- [14] H. Holloway. Registration error analysis for augmented reality, 1997.
- [15] Richard Holloway. *Registration Error in Augmented Reality Systems*. PhD thesis, University of North Carolina, Chapel Hill, North Carolina, 1995.
- [16] Douglas Kay. Transparency, refraction, and ray tracing for computer synthesized image. Master's thesis, Cornell University, Ithaca, New York, 1979.
- [17] Rudolf Kingslake. *Lens Design Fundamentals*. Academic Press, New York, 1978.
- [18] G. Klinker. Confluence of computer vision and interactive graphics for augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):433–451, 1997.
- [19] Craig Kolb, Don Mitchell, and Pat Hanrahan. A realistic camera model for computer graphics. *Computer Graphics*, 29(Annual Conference Series):317–324, 1995.
- [20] Bernard Kolman. *Elementary Linear Algebra*. Macmillan Publishing Co., Inc., New York, 1977.
- [21] Arthur Linksz. *Optics Volume 1: Physiology of the Eye*. Grune and Stratton, New York, 1950.
- [22] m. Bajura. *Mergin Real and Virtual Environments with Video See-Through Head-Mounted Displays*. PhD thesis, Univeristy of North Carolina at Chapel Hill, Chapel Hill, North Carolina, 1997.
- [23] H. Fuchs M. Bajura and R.Ohburchi. Merging virtual reality with the real world: Seeing ultrasound imagery within the patient. *IEEE Computer Graphics*, 26(2):203–210, 1992.
- [24] C. Maurer and J. Fitzpatrick. A review of medical image registration, 1993.
- [25] E. McGarrity and M. Tuceryan. A method for calibrating see-through head-mounted displays for augmented reality. *IEEE International Workshop on Augmented Reality*, October 1999.
- [26] Erin McGarrity. Evaluation of calibration for optical see-through augmented reality systems. Master's thesis, Michigan State University, East Lansing, Michigan, 2001.
- [27] James E. Melzer and Kirk Moffitt. *Head Mounted Displays: Designing for the User*. McGraw-Hill, New York, 1997.
- [28] P. Milgram and F. Kishino. A taxonomy of mixed reality visual displays. *IEICE Tranctions on Information Systems*, pages 1321–1329, 1994.
- [29] P. Min and H. Jense. Interactive stereoscopy optimization for head-mounted displays, 1994.

- [30] Gordon Moore. Cramming more components onto integrated circuits. *Electronics*, 38(8), Apr 1965.
- [31] Kenneth Ogle. *Optics: An Introduction for Ophthalmologists*. Charles C Thomas, Springfield, Illinois, 1968.
- [32] D. C. O'Shea. *Elements of Modern Optical Design*. John Wiley and Sons, 1985.
- [33] C. B. Owen, J. Zhou, K. H. Tang, and F. Xiao. Augmented imagery for digital video applications. *Handbook of Video Databases*, 2003.
- [34] R. Raskar, G. Welch, and W. Chen. Tabletop spatially augmented reality: Bringing physical models to life using projected imagery. *Second Int Workshop on Augmented Reality*, Oct 1999.
- [35] Ramesh Raskar and Kok-Lim Low. Interacting with spatially augmented reality, 2001.
- [36] G Reitmayr and D Schmalsteig. Location based application for mobile augmented reality.
- [37] Jannick P. Rolland and Terry Hopkins. A method of computational correction for optical distortion in head-mounted displays. Technical Report TR93-045, 1, 1993.
- [38] Robert Shannon. *The Art and Science of Optical Design*. Cambridge University Press, Cambridge, 1997.
- [39] Ivan Sutherland. The ultimate display. *Proceeding of IFIPS Congress*, 2:506–508, May 1965.
- [40] Ivan Sutherland. A head-mounted three-dimensional display. *AFIPS Conference Proceedings*, 33:757–764, 1968.
- [41] Kwok Hung Tang. Comparative effectiveness of augmented reality in object assembly. Master's thesis, Michigan State University, East Lansing, Michigan, 2001.
- [42] M. Tuceryan and N. Navab. Single point active alignment method (spaam) of optical see-through hmd calibration for ar. *Proceeding of the IEEE and ACM International Symposium of Augmented Reality*, 59:148–158, 2000.
- [43] A. Walther. *The Ray and Wave Theory of Lenses*. Cambridge University Press, New York, 1995.
- [44] Benjamin A. Watson and Larry F. Hodges. Using texture maps to correct for optical distortion in head-mounted displays. In *Proc IEEE Virtual Reality Annual International Symposium*, number 95-04, 1995.

- [45] Ross T. Whitaker, Chris Crampton, David E. Breen, Mihran Tuceryan, and Eric Rose. Object calibration for augmented reality. *Computer Graphics Forum*, 14(3):15–28, 1995.
- [46] Woojin Yu. Simulation of vision through actual human optical system. Master's thesis, University of California, Berkeley, Berkeley, California, 2001.
- [47] G. Zachmann. Distortion correction of magnetic fields for position tracking. In *Proc Computer Graphics International*, Belgium, Jun 1997. IEEE Computer Society Press.



MICHIGAN STATE UNIVERSITY LIBRARIES



3 1293 02460 4773