Interactive Graphical Model Building using Virtual Reality

by

Christopher Alexander Cooke

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

Bachelor of Science

and

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 1994

Author .....
Department of Electrical Engineering and Computer Science
May, 1994

Certified by .....
W. Eric L. Grimson
Associate Professor of Computer Science and Engineering
Thesis Supervisor

Certified by .....
Sharon Stansfield
Member of Technical Staff, Sandia National Laboratories
Thesis Supervisor

Accepted by .....
F. R. Morganthler
Chairman, Departmental Committee on Graduate Students

Interactive Graphical Model Building using Virtual Reality

by

Christopher Alexander Cooke

Submitted to the Department of Electrical Engineering and Computer Science
on May, 1994, in partial fulfillment of the
requirements for the degrees of
Bachelor of Science
and
Master of Science

Abstract

This paper presents a prototype system to create and verify computer generated graphical models of a remote physical environment. Virtual reality, specifically enhanced telepresence, is used to allow interaction between the user and the remote environment. A stereo view of the remote environment is produced by two CCD cameras. The cameras are mounted onto a robot which is slaved to a stereoscopic viewing device. This gives the user a sense of immersion in the physical environment. The stereo video is enhanced by overlaying the graphical models onto it. The overlay capability allows for visual verification of graphical models. Creation of a graphical model is composed of three steps: identification, marking, and placement of the object. The user is responsible for identification and marking of the object. Computer vision is used for the placement of the object.

Thesis Supervisor: W. Eric L. Grimson
Title: Associate Professor of Computer Science and Engineering

Thesis Supervisor: Sharon Stansfield
Title: Member of Technical Staff, Sandia National Laboratories

Acknowledgments

This work was performed at Sandia National Laboratories and was supported by the U.S. Department of Energy under Contract DE-AC04-94AL85000.

I would like to acknowledge Sharon Stansfield without whose help in the form of time, energy, and knowledge I would not have been able undertake this thesis. I would like to thank Professor Grimson for his patience and understanding. I would like to thank Larry Ray, Colin Selleck, and Chris Wilson who all provided vital assistance to my thesis. Thank you to Dave Strip for support and feedback on a weekly basis. A very special thank you to Nadine Miner. I would like to thank her not only for support and feedback, but also her friendship.

Thank you to Bobby Lai, my roommate of five years, for the laughing with me during the good times and supporting me through tough ones. To Jung-hua Kuo, I would like to say thanks for the memories. Mark Winters, thank you for the all your words of wisdom. Thank you to my friends and classmates at MIT that made college rich and fulfilling.

I am dedicating this thesis to my parents McHenry and Vivian Cooke. My parents have done so much for me, without their support this thesis would not be possible. Thank you Mom and Dad.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Cleanup of remote hazardous environments presents a challenge to the scientific community. Sandia National Laboratories (SNL) is attacking this problem with advanced telerobotic work. Their work is being directly applied to the remote retrieval of hazardous waste from underground storage tanks [6]. The tanks provide an opportunity to test geometric data collection techniques since partial or no *a priori* information is available about the environment inside these tanks. It is known that the tanks are filled with many obstacles such as cooling pipes, risers, and pumps. Since the sites are remote and dangerous, supervisory control of the robot is required. For a robot to navigate around these obstacles via motion planning, a complete geometric database of the remote site is needed. As a step toward being able to create a geometric database of remote sites, a prototype interactive graphical modelling system was constructed.

Graphical models present geometric information visually. This is important because humans can quickly process visual information, where as numbers and equations take longer to understand. Computer-aided design (CAD) packages transform geometric data to a graphical form. The drawback of CAD packages is that geometric data is hand entered by the user. Data entry is both time consuming and prone to error. Another problem is that the CAD model cannot be easily verified against the site it is modelling. To alleviate these problems this thesis explores the combination of a CAD package and computer vision system. The computer vision system is unique

in that a human operator assists in the process of object recognition and scene segmentation. The assistance of a human operator is made possible by the use of virtual reality equipment.

The thesis is broken down in the following manner. Chapter two describes the overall system. The chapter includes both the hardware and software used. Chapter three describes the computer vision system. This includes camera and robot calibration methods, as well as computer vision techniques used to extract geometric information. Chapter four describes the experiments used to test the system. Chapter five concludes with a description of future work.

The problem that this thesis addresses is the collection and verification of geometric information from a remote site. The goal is to build an interactive graphical modelling system (IGMS) that is capable of collecting geometric data and allows visual verification of that data.

Collection of geometric or depth information can be carried out by active or passive sensors. Active sensors send out packets of energy. A calculation is performed on time of flight of the energy packet to determine the depth information. The wavelength of the energy packet determines the accuracy that can be achieved using an active sensor. Passive sensors absorb energy from the environment. The correlation of the energy absorbed from more than one location provides the depth information.

Both active and passive sensors are used in this thesis to extract depth information. Ultrasound (active sensor) is used as a first pass at the depth of objects. Two CCD cameras (passive sensors) are used to refine the locations of objects.

The presentation of raw geometric data points makes visual verification difficult. To improve the verification process the geometric information collected is presented in the form of objects (e.g. block, cylinder). Objects do not overload CAD packages as fast as raw depth information would. Objects also allow faster motion planning for robots.

Object recognition by a computer vision system is an active area of research. To avoid the difficulties of this problem, a human operator is introduced into the system to recognize objects of importance. The need for an intuitive interface between the

computer vision system and the human operator brought about the introduction of virtual reality (VR) equipment.

VR provides visual information to the user through a stereoscopic viewing device. The two CCD cameras, located at the remote environment, provide the visual information displayed in the stereoscopic viewer. By slaving the motion of the camera platform to the motion of the stereoscopic viewer, the user is given a sense of immersion in the remote environment. The stereo graphics of the remote site are provided by the CAD package. By registering or matching the graphical cameras with the stereo camera system the merging of reality with virtual reality is possible. This technique is referred to as enhanced telepresence. In enhanced telepresence, the real environment is enhanced by overlaying graphical information onto it.

Virtual tools are devices that exist in the graphical environment. They are used to assist the operator of the system. A 3D pointer, a virtual tool, is used to allow the user to interact with modelling system. The 3D pointer marks points of interest in the remote environment. These points of interest are then used by the modelling system to create graphical objects.

To summarize, the interactive graphical modelling system (IGMS) uses enhanced telepresence to allow the operator to interact with the computer vision system. Objects are identified by the operator; the computer vision system then extracts depth information about the object and creates a graphical model. The location and dimension of the graphical object can then be visually verified by user.

1.1 Previous Work

Work done at the GE Advanced Technology Laboratories shows that a complex geometric database can be verified and maintained by using graphical overlays onto stereo video [12]. Verification is a visual process carried out by the operator to confirm that wireframe models are properly positioned with respect to physical objects. Verification is required for telerobotic operation in which the geometric database is going to be used for motion planning. Maintenance allows the graphical objects that are cur-

rently in the database to be moved by the operator to match their physical position. Maintenance provides a method for updating inaccurate models so that a telerobotic operation may be started or completed. For example, if a robot moves an object, maintenance needs to be performed on the geometric data to keep it up-to-date.

At the University of Toronto, Paul Milgram and David Drascic have merged stereoscopic video with stereoscopic computer graphics to produce a *Virtual Tape Measure* [11]. The *Virtual Tape Measure* used a graphical 3D pointer to make measurements in a physical scene. Both relative distance and absolute distance measurements could be made with the pointer.

The work done at GE Advanced Technology Laboratories and the University of Toronto demonstrated several important points.

- Manipulation of graphical objects in a physical environment is useful.
- An operator can make relative position judgments with both real and virtual objects.
- It is possible to make physical measurements from points created by a virtual tool.

Bruce Bon and colleagues at Jet Propulsion Laboratory designed and built a prototype telerobotic system for graphical model building called *Operator Coached Machine Vision* (OCMV) [4]. The operator of OCMV indicates edges on a physical object by controlling a 3D virtual pointer with a joystick. When the operator has indicated all the edges of the object, machine vision takes over and resolves the edge more accurately than the operator could indicate. Once the edges are resolved the computer creates a wireframe model of the object. The machine vision system took inputs from four camera positions. The user interface was flat screen, making immersion in the remote environment impossible.

The difference between the research in this thesis and the work mentioned above is immersion of the operator in the remote environment. Immersion is done by slaving the motion of the camera platform to the motion of the viewer. Also, interaction between the IGMS and user is based on a VR paradigm. One example of this is that voice commands are used as input. Another difference is that texture mapping

of graphical objects is utilized to create more realistic looking models. The texture maps also assist in identification of objects. The computer vision system uses only two cameras to extract dimensional information from the remote environment and relies heavily on operator assistance.

Chapter 2

System

This chapter describes the interactive graphical modelling system (IGMS) developed at SNL. The chapter starts with an overview of the system's function. Then a description of the hardware and software is given. The chapter closes with a high level description of the custom software used to control the IGMS.

The interactive graphical modelling system (IGMS) is composed of three major components, a CAD package, a computer vision system, and a VR user interface. The CAD package is used for both its graphics display and geometric database. The computer vision system is used to extract geometric data from the remote site. The VR equipment provides natural control and feedback to the user. The three components are connected together with custom software. Figure 2-1 illustrates how the IGMS is organized.

The IGMS is designed to build and verify simple graphical models of a remote site. Verification means that graphical models are visually confirmed by the operator to see that they match with respect to size and location. Two viewing modes, solid and wireframe, exist. The different modes enhance the ability to verify objects. Models are constructed in the graphical environment using position and dimensions obtained from the computer vision system. The IGMS has two basic models, cylinder and block, which are scaled to match objects in the remote environment. A cylinder was chosen as a primary model because the underground storage tanks have a large number of pipes in them. Block was selected as the other primitive so that objects

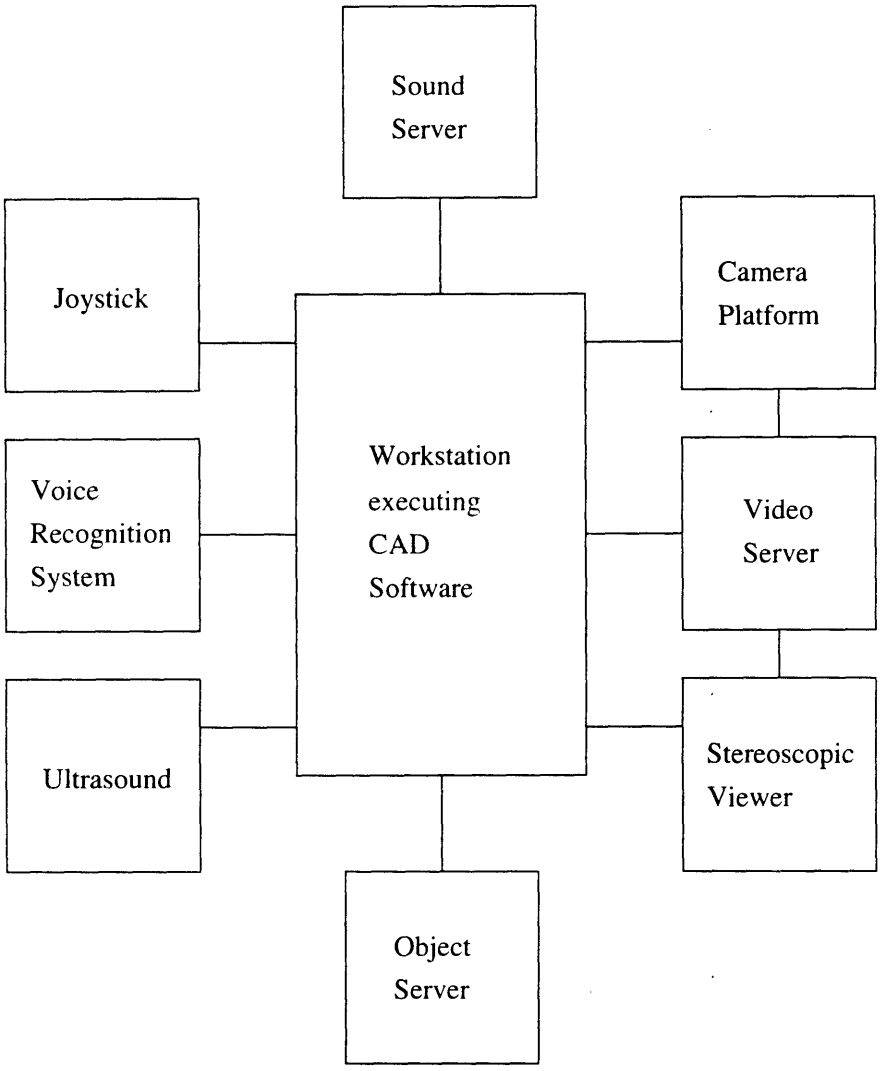

Figure 2-1: Interactive Graphical Modelling System Block Diagram

of interest could be bounded. To make up for the crudity of the block model, the capability of texture mapping blocks is available. Texture maps are pictures of the actual object obtained from the digitized image. These pictures are pasted onto the block model.

The IGMS is based on a VR user interface. The VR user interface provides visual and auditory feedback. Visual feedback is displayed in the stereoscopic viewer. Visual information is received from two sources: the cameras located at the remote site and the graphical cameras created by the CAD package. Special hardware that uses chroma keying overlays the stereo graphics onto the stereo video. A camera platform orients the cameras to match the orientation of the stereoscopic viewer.

Auditory feedback is played over speakers. It is used to inform the user about the current state of the IGMS.

The VR interface receives input from verbal commands and both head and hand motion of the operator. The stereoscopic viewer tracks the head motion. A joystick tracks hand motion. A voice recognition system interprets verbal commands.

The computer vision system is based upon the concept of stereo vision. Two monochrome CCD cameras provide the stereo view. A frame grabber captures images from both cameras. Edge detection and scene analysis are performed in software.

2.1 System Hardware

The system hardware has a workstation as the central processing unit. Information is collected from several peripheral devices by serial communication. Other hardware in the system is accessed over the network. Below is a description of the system hardware used.

2.1.1 Workstation

A Silicon Graphics Crimson^{TM} with Reality Engine^{TM} Graphics is used to run to CAD software. The Reality Engine^{TM} Graphics allow texture mapping to be performed in hardware. This makes the texture mapping process real time.

2.1.2 Polled Devices

The stereoscopic viewer, joystick, ultrasound, and voice system are all polled devices. Polling means that information is requested from these devices by the workstation from a serial data port. Except for the ultrasound, all the devices are polled before each update of the computer graphics. The ultrasound is only polled upon an operator request. To increase the speed of polling these devices a request for data is made one cycle before it is read. This gives a faster response time for the IGMS, but also introduces data latency. Since the graphics are being updated at approximately 30 frames a second, data latency does not present a problem. To implement this polling method, the software provided by the manufacturers of the different devices was modified before it was incorporated into the control software. Below is a description of each of the polled devices along with the functions they perform in the system.

2.1.3 Stereoscopic Viewer

The Fakespace BOOM2C^{TM} is a six degree of freedom, stereoscopic display device. Figure 2-2 shows a picture of the Fakespace BOOM2C^{TM}. It is mechanically tracked, producing both low latency and high accuracy. It transmits its position and orientation information over a serial line at 9600 baud using a compact protocol. The maximum polling rate of the Fakespace BOOM2C^{TM} is $70Hz$. The mechanical tracking is relative to the power up position; this means that the position of the Fakespace BOOM2C^{TM} at power up is considered the user-defined zero position. The resolution of the stereoscopic viewer, which is CRT based, is 1280x1024 pixels per channel. Fakespace Inc. provides a technique which allows the production of a stereo graphics from one workstation to be displayed in the viewer. This technique was incorporated into CAD software.

Two buttons are located on the handle of the Fakespace BOOM2C^{TM}. The condition of these buttons is also transmitted on a data request. These buttons are mapped to IGMS commands. Table 2.1 shows the different commands the two buttons execute. In the table, one means that the button is being pressed; zero means that the

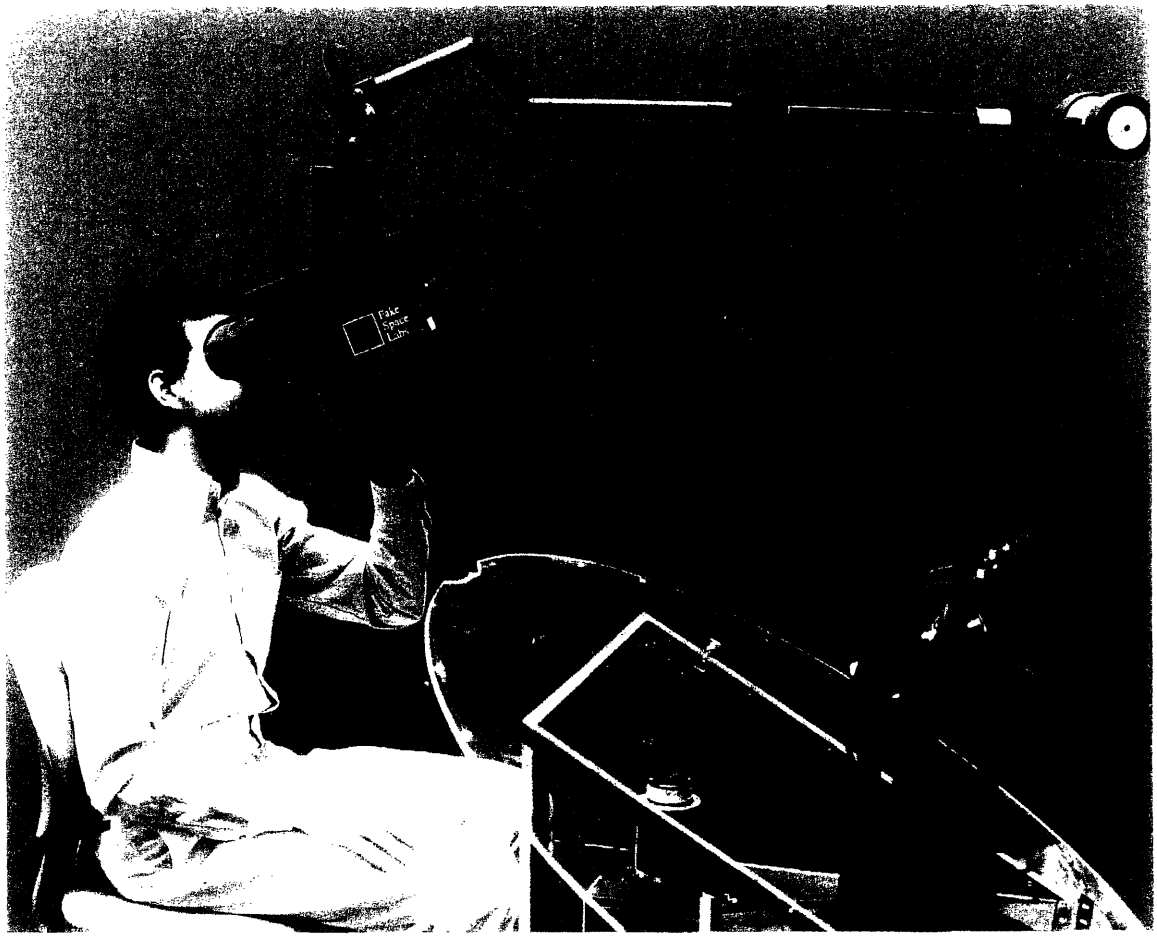

Figure 2-2: Fakespace BOOM2C^{TM}

Buttons	*Value*	*System Control*
00	0	None
01	1	Freeze Graphics and camera platform
10	2	Ultra Sound Ranging
11	3	Slave Graphics and camera platform to Viewer

Table 2.1: Mapping for Fakespace BOOM2C^{TM} Buttons in the IGMS

button is not being pressed.

The Fakespace BOOM2C^{TM} is used by the IGMS to give the user a sense of immersion in the environment. The fact that the Fakespace BOOM2C^{TM} is capable of tracking the pose of the viewing location makes immersion possible. Stereoscopic viewing devices that do not track the viewing position do not allow for immersion (e.g. flat screen display).

2.1.4 Joystick

The FlyBox^{TM}, manufactured by BG Systems, is composed of a three degree of freedom joystick, two levers, and several switches. It communicates with the workstation over a serial line at 9600 baud. Its maximum polling rate is $50Hz$. The communication protocol is decoded so that the switches return a value of 1 if on and 0 if off. The joystick is used to control the graphical pointer. Movement of the joystick returns a value between 1 and -1 for each axis. Table 2.2 summarizes the functions of the FlyBox^{TM} in the IGMS. Currently the levers are not used by the IGMS.

2.1.5 Ultrasound

A Contaq UDM-FL^{TM} Ultrasonic Transducer was used for ultrasonic ranging. The sensor has a range from two inches to sixty feet with a beam width of twelve degrees. It communicates with the workstation over a serial line at 9600 baud. The maximum polling rate of the sensor is 6 hertz. This slow update means that the IGMS graphics would be slowed down if it were polled every display cycle. Therefore, it is only polled

Action	*Value*	*Mapping*
Forward/Backward	(1 -1)	cm per frame
Left/Right	(1 -1)	cm per frame
Twist	(1 -1)	cm per frame
Button 1	[1, 0]	removing point
Button 2	[1, 0]	Wirefame / Solid
Button 3	[1, 0]	removing model
Button 4-8	[1, 0]	NA
Trigger	[1, 0]	Mark point
Slider 0,1	(1 -1)	NA

Table 2.2: Mapping for the FlyBox^{TM} in IGMS

Ranging Distance	*Beam Width*
0.5 m	0.104 m
1.0 m	0.208 m
2.0 m	0.416 m
5.0 m	1.040 m

Table 2.3: Ultrasound Ranging

upon a user request.

With the current implementation of the IGMS, it was found that many users have trouble with depth perception. The problem is caused by monocular depth cues. With the overlay technique being used graphical objects are still visible when they are placed behind real objects. This tends to confuse novice users. If a graphical object is placed far enough behind a real object, the user can no longer converge the graphics. This prevents the user from placing the pointer far behind an object. To help users with depth perception, ultrasonic ranging was introduced into the IGMS.

Table 2.3 shows that as object distance increases ultrasonic ranging of the correct object becomes more difficult due to the beam spread. If an object is surrounded by other objects, depth adjustment of the graphical pointer will be required by the operator.

Command	*Value*	*Function*
Freeze	1	Freezes graphics and video movements
Range	2	Uses ultra sound to range the depth of object
Unfreeze	3	Slaves the graphics and the video to the viewer
Box	10	Informs the system to model the object as a box
Cylinder	11	Informs the system to model the object as a cylinder

Table 2.4: Verbal Commands

2.1.6 Voice Recognition System

Dragon Writer^{TM}, a product by Dragon Systems Inc., was used for speech recognition. The voice recognition hardware was installed in an IBM compatible personal computer. It allows users to design a vocabulary to be recognized. The vocabulary is created from a user defined language. A language may be a list of words or a complete set of sentence structures. With software written at SNL, the speech recognition system on the PC communicates with the workstation over a serial line at 9600 baud. This software sends over two ascii digits whenever a verbal command is recognized.

The vocabulary for the system was composed of the following words: freeze, unfreeze, range, cylinder, and box. Dragon Systems uses a finite state machine to represent the language. Each state shows the valid words at that point. Figure 2-3 show the finite state machine for the language used. Table 2.4 states the function of each verbal command.

Since no command has a unique origin, the state of the voice system can fall behind the state of the IGMS. In other words, if the IGMS receives a command from the Fakespace BOOM2C^{TM} buttons the state of the voice system may be incorrect. Currently, the workstation is unable to update the state of the voice recognition system, so if an operator freezes the graphics using the buttons, the verbal command unfreeze would not be valid. This forces the operator to unfreeze the graphics with the buttons or to manually update the voice recognition by saying freeze.

The voice recognition system may be set up so that all words are valid to overcome the above problem, but this decreases the recognition rate. Also, the more words

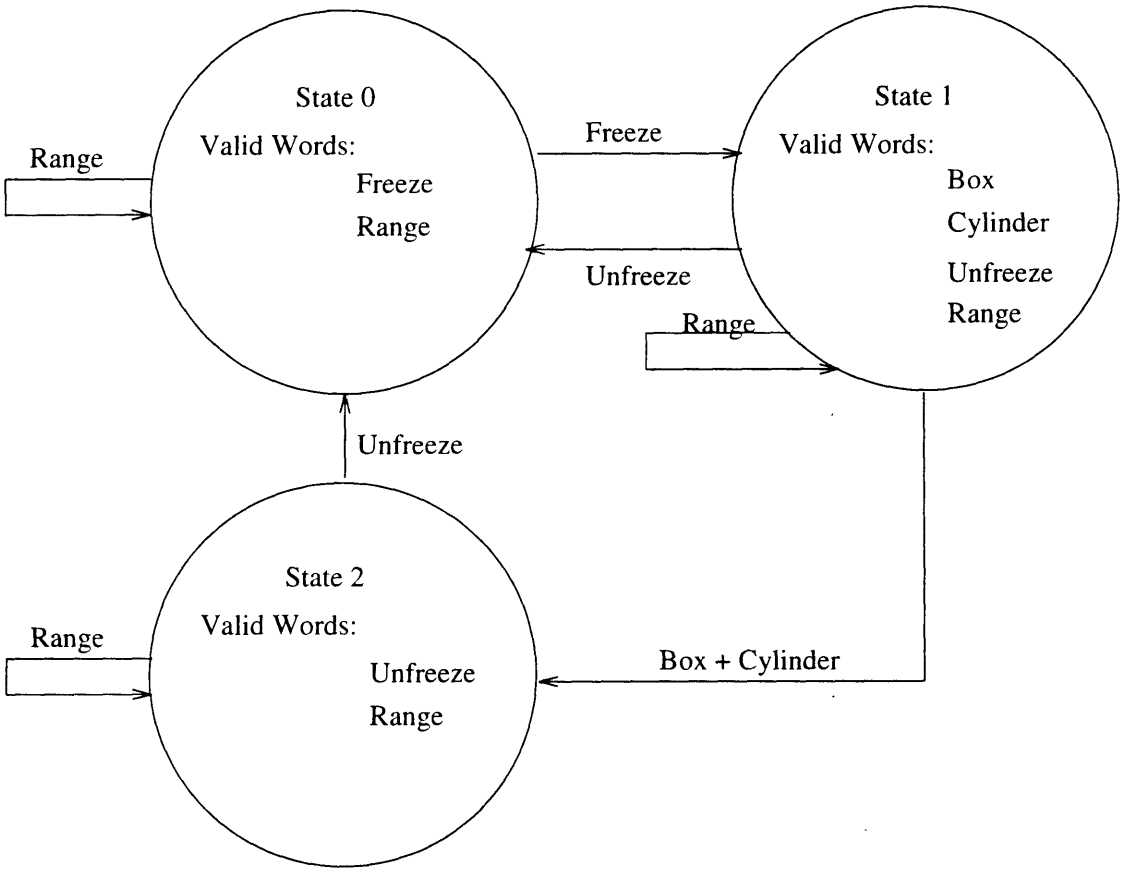

Figure 2-3: Voice Recognition Finite State Machine

available the more likely random speech will be interpreted as a vocal command.

The vocabulary was taught to the voice recognition system. It was trained with five male voices and five female voices in an attempt to create a speaker independent system.

2.1.7 Camera Platform

The Fakespace MOLLY^{TM} is a three degree of freedom robot with an adjustable camera platform. Figure 2-4 shows a picture of the Fakespace MOLLY^{TM}. MOLLY^{TM} communicates with the workstation over a serial link at 9600 baud. A personal computer is normally used to slave MOLLY^{TM} to the stereoscopic viewer. Since the IGMS needs to control the camera platform independent of the stereoscopic viewer, MOLLY^{TM} was directly interfaced to the workstation. MOLLY^{TM} is different from the other polled devices because its position may be both set and requested from the workstation. The need to request the position of MOLLY^{TM} is necessary because it may not be able to move to the position set by the workstation.

The interface provides direct control over the robot's specified orientation. Control of the speed of the robot is accomplished by setting the number of steps per move. Also, MOLLY^{TM} may be queried as to its current orientation. The power up position is the zero or home position of the robot. The current orientation can be reset to zero.

2.2 Server/Client

A server/client setup is usually used for high speed communication between workstations. To accomplish this the server and client communicate with each other over a network. The server/client setup is used to offload computing cycles or to obtain access to equipment on other workstations in the IGMS. Software was written to set up both server and client workstations on the network. The sound server and the video server are both utilized to gain access to specialized equipment. An object server is used to separate the system control software from the computer vision software. This

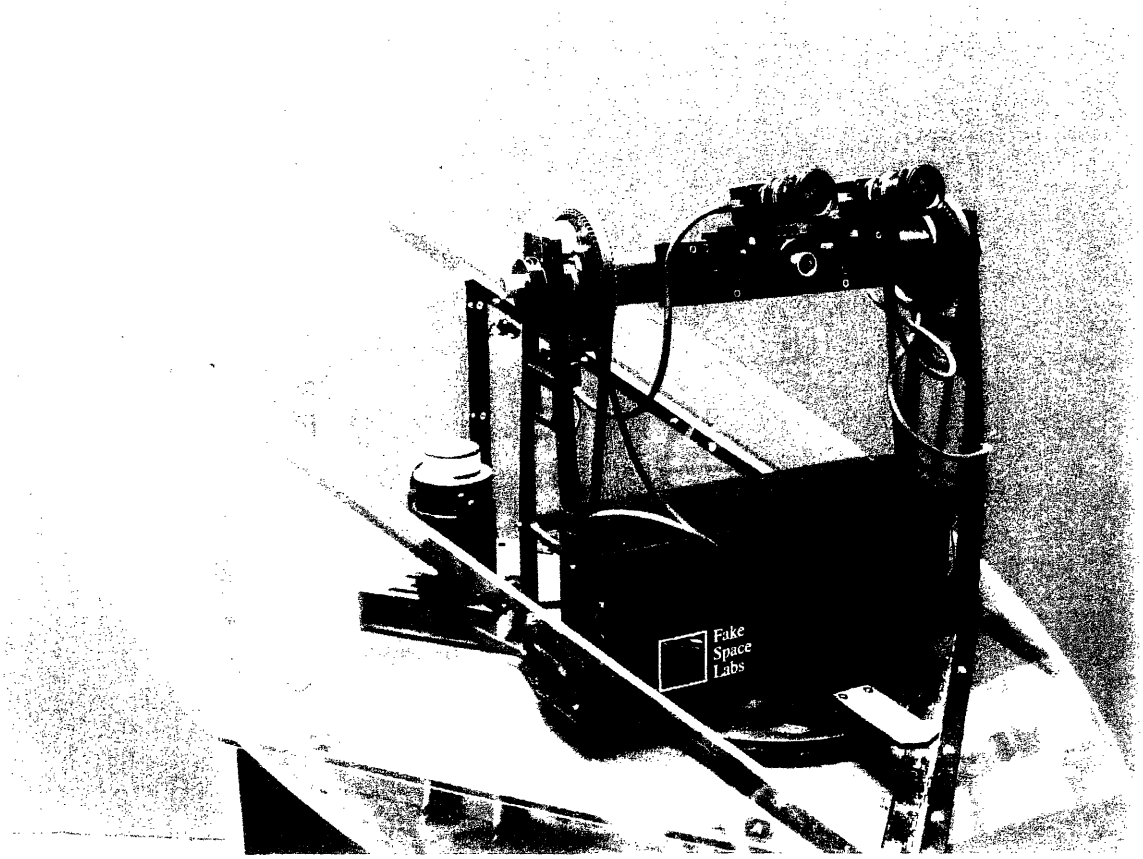

Figure 2-4: Fakespace MOLLY^{TM}

Sound	*Feedback*
Camera Click	Informs the user that the graphics is freezing.
Ricochet	Informs the user that ranging is taking place.
Woodblock Hit	Informs the user that a mark has been placed.
Belltree	Informs the user that a new graphical object has been added.
Introduction music	Used to grab the user attention.

Table 2.5: Sound Feedback for IGMS

allowed the two pieces of software to be tested independently.

2.2.1 Sound Feedback Server

A sound server written at SNL uses the sound board of an SGI Indigo^{TM} to play back prerecorded messages and sounds. Sounds are digitally recorded and stored in files. Access and playback of the sound files is made possible by the sound server. Each file has a number id. When the sound server receives an identification number it plays the corresponding sound over the speakers.

Tables 2.5 and 2.6 show both sound and verbal messages used by the IGMS. Most of the sounds and messages are used as feedback to the user, but one of the messages is used to query the user for input.

2.2.2 Video Server

A video server was written to control the overlaying of the graphics onto the video. The server uses a RGB/VIEW^{TM} model 1050 to control the overlay process. The RGB/VIEW^{TM} allows for low resolution video to be overlaid onto high resolution video. It is also used to acquire low resolution video frames. It takes approximately three seconds to store a frame of video. Once a video frame is stored, the video server performs edge detection on the video frame. It takes approximately fourteen seconds to perform the edge detection algorithm on one video image. The edge detection algorithm used was developed by John Canny and is explained in chapter 3. The

Verbal Message	*Meaning*
Modeling as (box, cylinder).	Informs the user of the type of model being created.
Freezing viewing position.	Informs the user that graphics are not slave to the viewer
Unfreezing viewing position.	Informs the user that graphics are slaved to the viewer.
Removing (points, model).	Informs the user that a virtual object is being removed.
Please wait.	Lets the user know that the computer is doing calculations.
Locating object.	Informs the user of the type of calculation taking place.
Welcome to the virtual reality lab.	Introduction Message
This a an interactive computer vision demonstration.	Introduction Message
Is object a box or a cylinder?	Asks the user for identification of the physical object

Table 2.6: Verbal Feedback for IGMS

reason that edge detection is performed by the video server is that while the operator is marking an object the edge detection can be taking place on another machine.

The RGB/VIEW^{TM} video interface is used to overlay the stereo graphics onto the stereo video. Low resolution (512×480), live stereo video is produced by two monochrome CCD cameras. The workstation produces a high resolution (1280×1024) RGB video signal. Thus, the RGB/VIEW^{TM} must enlarge the low resolution video to match the high resolution produced by the workstation.

The video interface allows the user to select a color level for each channel (Red, Green, and Blue) in which the the graphics will be overlayed onto the graphics. The video interface breaks each channel into 256 different level or shades. The current setup allows all shades above sixty to be overlayed onto the stereo video. In other words, wherever the graphics are black the camera signal will be seen. This means that a wireframe model for floors and walls must be used, or else the stereo video would not be visible. In other words, solid models block the stereo video in those

image points.

The video interface allows the user to move the stereo video around in the stereo graphics by specifying the pixel offset in u and v direction. This ability makes it easier to set up the registration of the graphics with the video.

2.2.3 Object Server

An object server was created to locate objects in the video. The server takes the type of object and the pixel information from the system control software. It combines this information with the images from the video server and returns the pose and dimension of the object. More detail about the object server is provided in chapter 3.

2.3 CAD Software

The CAD package used is SILMA's *CimStation*^{TM}. *CimStation*^{TM} is a robotics simulation software package which can be customized to allow added functionality. Customization software was written in SIL. SIL is an object-oriented language that *CimStation*^{TM} provides for programming. Also, C and Fortran code can be linked into *CimStation*^{TM}. Using SIL, graphical objects (eg. block, cylinder, sphere) can be created and positioned within a workcell during execution. SIL code provides direct control over the position and properties of the graphical cameras.

To set up the graphical cameras *CimStation*^{TM} needs five parameters: field of view, aspect ratio, front clipping plane, back clipping plane, and pose. Since these parameters are all setable in software, it is possible to match the virtual cameras with the real cameras. *CimStation*^{TM} was selected as the CAD package to use because of its flexibility. This flexibility made system integration possible.

To customize *CimStation*^{TM} for the IGMS several extensions were added. The interfaces to all the hardware devices were written in C and linked into *CimStation*^{TM}. The IGMS control software was written in SIL. This software controls the graphical environment within *CimStation*^{TM}, as well as the IGMS hardware. Code was written

in C to access the network so that *CimStation*^{TM} could access the sound, video, and object servers.

2.4 Control

The control software for the IGMS is executed from inside the CAD package. There are two modes of operation, verification and model building. Below is a description of the two modes of operation and the initialization process. Figure 2-5 shows the key features for the two modes of operation.

On power up, the control software goes through an initialization process. First, it sends requests to attach to the sound, video, and object servers, then it loads calibration data for the cameras and the camera platform. It clears the serial lines and requests data from each device. If any of the steps are not completed successfully the control software will exit. Otherwise, verification mode is entered.

In verification mode, the camera system is slaved to the stereoscopic viewer. The pose of the viewer is requested. From this information, the orientation of the camera platform is updated to match the orientation of the viewer. Also, the pose of the graphical cameras is updated. The pose of the graphical cameras is computed by using the calibration data for the camera platform. The voice recognition system and joystick are polled. If the freeze command is issued by the user through a voice command or button press the IGMS will enter model building mode. Otherwise the verification process will be repeated, starting with polling the position of the stereoscopic viewer.

By slaving both the graphical and the real cameras to the stereoscopic viewer, verification can be visually accomplished by examining the remote environment. Having both solid and wireframe models available allows the user to check how well the graphical models match with the remote environment. The voice command "wireframe" displays the wireframe or outline model of the graphical objects. This feature is necessary since graphical overlay with solid models can hide model mismatch. This happens when the graphical model is larger than the object it is modelling.

When any command is detected, sound feedback is provided to the user by the sound server.

In model building mode, the camera position is fixed. Fixing the camera position allows the image to be stored in memory before the object to be modelled is completely identified. Upon entering model building mode a request is sent to the video server to grab an image from the two CCD cameras. These images will be used by the object server to return the dimensions of the object being graphically modelled. A graphical 3D pointer is placed into the scene. Knowing the orientation of the camera platform allows for the placement of the pointer. On every update cycle the voice system, stereoscopic viewer, and joystick are polled to look for commands being issued. The pointer is controlled by the joystick or the ultrasound system. To get a rough estimate of the depth of the objects the "range" command is issued. This command moves the graphical pointer to the depth returned by the ultrasound system. Since the ultrasound sensor is mounted between the cameras, the depth information should correspond to the object of interest. By moving the pointer with the joystick, four points can be placed that bound the object. The location of these points will be sent to object server. A point is made by issuing the "mark" command. Points may also be removed by issuing the "unmark" command. Sound and visual feedback inform the user when a point is marked or unmarked.

Once the four marks have been placed the system prompts the user to identify the type of object that is to be modelled. This information is sent off to the object server and the system waits for geometric data about the object to be returned. If the object server cannot locate the object it returns a negative value, otherwise it returns the dimensions and location of the object. The control system then enters the geometric data into the CAD package and a graphical object is created.

To enter the verification mode from modelling mode the "unfreeze" command is issued. Otherwise, the computer awaits a new set of bounding points.

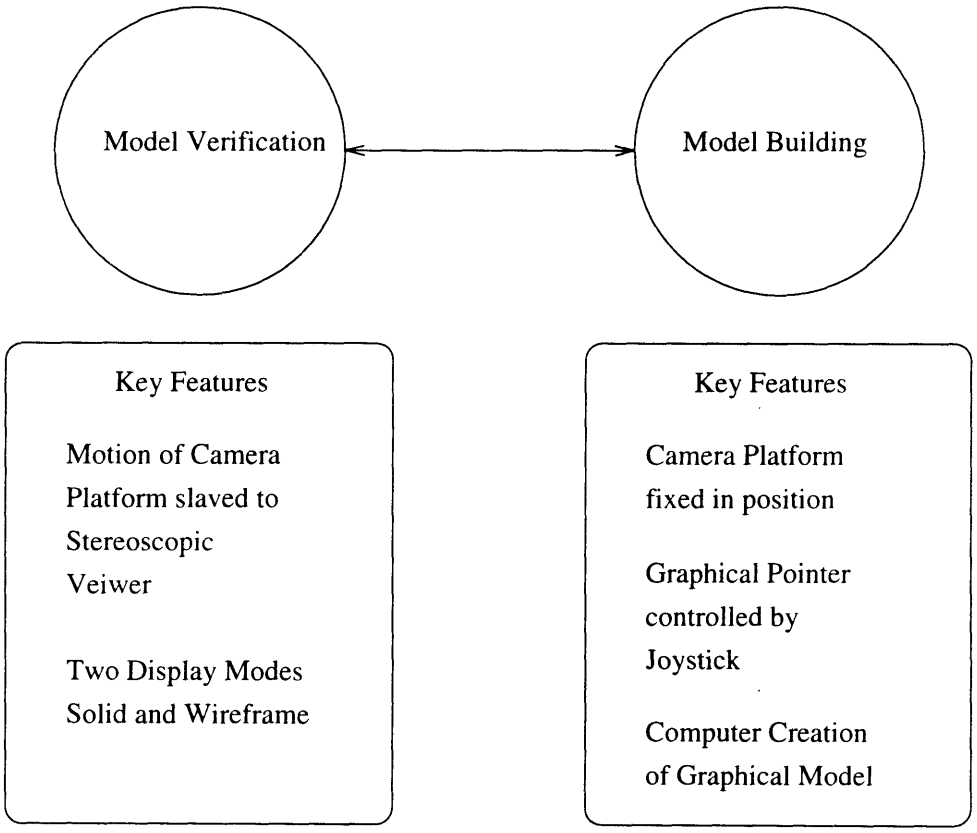

Figure 2-5: Two Modes of IGMS Control Software

Chapter 3

Computer Vision System

In this chapter, a description of the computer vision system is presented. The chapter begins with the camera model used and a method for calculating it is given. The image processing software is outlined. Lastly, the steps involved in moving from images to objects is shown.

The computer vision system is composed of several components: two monochrome CCD cameras, a camera platform, a framegrabber, and image processing software. Models for both the cameras and the camera platform are created so that they may be accurately represented in the IGMS software. The camera model is based on a point projection model. The camera platform model uses the *Denavit-Hartenberg* notation to calculate a transformation matrix that relates the position of the cameras to the base position of the camera platform.

The models provide a method to relate the graphical environment to the real environment. The camera model supplies the camera parameters needed to match the graphical cameras to the real cameras. To obtain the parameter values the camera model is decomposed into its component parts. The camera platform model furnishes the pose of the cameras when the platform is moving. If the cameras are fixed in position then the camera model alone provides the pose.

A stereo camera system is used to produce a 3D image. A line projecting from each camera is used to triangulate the position of points in world space. This line is calculated using inverse perspective. The 3D points are combined to create graphical

objects that are correctly positioned and have the proper dimensions in the graphical environment with respect to the remote environment. In this chapter, the above models and concepts will be developed.

3.1 Camera Model

A camera model transforms three dimensional world points to two dimensional image points. Point projection is used to model the camera. In the point projection model, an image is obtained by projecting the scene through a single point (point of focus) onto an image plane. The distance between the point of focus and image plane is referred to as the focal length. Using geometry, it can be shown that the focal length is one of the scaling factors between the actual size of an object and the size of the object in the image plane. The camera calibration matrix explained below is used to encapsulate the point projection camera model. The point projection model makes no attempt to adjust for lens distortion.

One of the tools used in computer vision and computer graphics is homogeneous coordinates. Homogeneous coordinates are used to create linear transformation matrices. To convert a physical point (x,y,z) to a homogeneous coordinate, the point is scaled by an arbitrary nonzero scale factor k. The new coordinate is (kx, ky, kz, k). This transforms a 3D point to a line. Hence, a single point has an infinite set of values in homogeneous coordinates. This representation of points will be used in the derivation of the camera calibration matrix.

3.2 Camera Calibration

The camera calibration used is outlined in Ballard and Brown[2]. Calculation of the camera calibration matrix is based on the least squares solution for an overdetermined system of linear equations. The camera calibration matrix is a 3×4 matrix. The matrix transforms homogeneous world coordinates into homogeneous image coordinates. This is shown by equation 3.1.

$$[C] \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} u \\ v \\ t \end{bmatrix} \tag{3.1}$$

To transform the image point to a homogeneous coordinate system ($(U, V) \Rightarrow (u, v, t)$) equation 3.2 and 3.3 are applied. By selecting a scale factor of one for the world points the transformation is simplified ($X = x$, $Y = y$, and $Z = z$).

$$U = \frac{u}{t} \tag{3.2}$$

$$V = \frac{v}{t} \tag{3.3}$$

The matrix C has 12 elements. Since a homogeneous coordinate system is being used, the scaling element C_{34} is arbitrarily set equal to one. Now, 11 unknowns exist. To form an overdetermined system at least 12 equations are needed. It will be shown below that each data point (a world point and corresponding image point) provides two equations, therefore six data points or more are needed to calculate the camera calibration matrix.

The matrix setup used to find the least squares solution is given in equation 3.4. The matrix A contains only constants. The vector x contains all the unknown variables and the vector b is another set of constants. The error is $E = |Ax - b|$. It can be shown that the square error is minimized when $A^T Ax = A^T b$ [14]. Hence, $x = (A^T A)^{-1} A^T b$ is the solution to the problem.

$$A \quad x \quad = \quad b \tag{3.4}$$

Transforming equation 3.1 into the least square format is explained below. The vector x is equal to the unknown elements of the matrix C. The constants are provided by the data points collected. Rewriting the matrix equation 3.1 into a set of linear equations gives equations for u, v, and t (3.5 - 3.7).

$$u = xC_{11} + yC_{12} + zC_{13} + C_{14} \tag{3.5}$$

$$v = xC_{21} + yC_{22} + zC_{23} + C_{24} \tag{3.6}$$

$$t = xC_{31} + yC_{32} + zC_{33} + 1 \tag{3.7}$$

Rewriting equations 3.2 and 3.3 gives the relationship shown in equations 3.8 and 3.9.

$$u - Ut = 0 \tag{3.8}$$

$$v - Vt = 0 \tag{3.9}$$

Substituting for u, v, and t the following relationships are established.

$$[xC_{11} + yC_{12} + z_{13} + C_{14}] - U[x_{31} + yC_{32} + zC_{33} + 1] = 0 \tag{3.10}$$

$$[xC_{21} + yC_{22} + zC_{23} + C_{24}] - V[x_{31} + yC_{32} + zC_{33} + 1] = 0 \tag{3.11}$$

The linear equations 3.10 and 3.11 are transformed to the least squares format. This is shown by the matrix equation 3.9 where x_i is the x location for the i data point collected, y_i is the y location for the i data pointed collected, and so on.

$$\begin{bmatrix} x_1 & y_1 & z_1 & 1 & 0 & 0 & 0 & 0 & -U_1x_1 & -U_1y_1 & -U_1z_1 \\ 0 & 0 & 0 & 0 & x_1 & y_1 & z_1 & 1 & -U_1x_1 & -U_1y_1 & -U_1z_1 \\ x_2 & y_2 & z_2 & \cdot & \cdot & \cdot & & & & & \cdot \\ \cdot & & & & & & & & & & \cdot \\ \cdot & & & & & & & & & & \cdot \\ \cdot & & & & & & & & & & \cdot \\ 0 & 0 & 0 & 0 & x_n & y_n & z_n & 1 & -U_nx_n & -U_ny_n & -U_nz_n \end{bmatrix} \begin{bmatrix} C_{11} \\ C_{12} \\ \cdot \\ \cdot \\ \cdot \\ \\ C_{33} \end{bmatrix} = \begin{bmatrix} U_1 \\ V_1 \\ \cdot \\ \cdot \\ \cdot \\ U_n \\ V_n \end{bmatrix} \tag{3.12}$$

After six or more data points are collected, matrix equation 3.12 is used to calculate the camera calibration matrix.

3.2.1 Collection of Data Points

Data points for the camera calibration are collected by grabbing images of a ruler placed on the optical table. Software was written to allow points in an image to be selected using a mouse. The u,v values are manually collected using this software. The x,y,z position is measured using the ruler and the optical table. These data points are then entered into the camera calibration software written by Larry Ray, a member of the technical staff at SNL. The software performs the calculation outlined above.

3.3 Decomposition

Once a camera has been calibrated using the above method, the camera calibration matrix is decomposed. The method for decomposition of a camera calibration matrix is taken from Ganapathy [7]. The decomposition provides the image center (u_0, v_0), scale factors (k_u, k_v) from image units (pixels) to world units (inches), and the pose of the camera center in terms of the world coordinate system. From this, field of view and the aspect ratio can be derived.

The first step is to identify the matrices that make up the calibration matrix C . Below is a step by step construction of camera calibration matrix.

First the origin must be moved to the position of the camera center (X_c, Y_c, Z_c). A translation matrix is used to do this and is given by equation 3.13.

$$T = \begin{bmatrix} 1 & 0 & 0 & -X_c \\ 0 & 1 & 0 & -Y_c \\ 0 & 0 & 1 & -Z_c \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.13}$$

Next the X-Y plane is rotated about the Z axis until the Y axis is parallel to the ray pointing from the camera center to the center of the image plane. Matrix R_θ is used for rotation about the Z axis.

$$R_\theta = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.14}$$

The Y axis is aligned with the ray pointing from the camera center to the origin of the image plane by rotation about the X axis. Matrix R_ϕ is used for rotation about the X axis.

$$R_\phi = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\phi & \sin\phi & 0 \\ 0 & -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.15}$$

To align the X axis with the u axis of the image plane, the X-Z plane is rotated about the Y axis. Matrix R_ψ is used for rotation about the Y axis.

$$R_\psi = \begin{bmatrix} \cos\psi & 0 & -\sin\psi & 0 \\ 0 & 1 & 0 & 0 \\ \sin\psi & 0 & \cos\psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.16}$$

Combining these four matrices gives a new coordinate system that is with respect to the camera center and image plane. Ganapathy calls the resultant coordinate system given by equation 3.17 the image centered coordinate system.

$$W = R_\psi R_\phi R_\theta T \tag{3.17}$$

New 3D coordinate values are computed with respect to matrix W. This is shown by equation 3.18.

$$[W] \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} \tag{3.18}$$

To adjust for the properties of the camera another matrix needs to be applied. A perspective matrix is applied to scale 3D coordinates onto the image plane.

$$x'' = \frac{x'f}{y'} \tag{3.19}$$

$$z'' = \frac{z'f}{y'} \tag{3.20}$$

The coordinates on the image plane must be scaled to the size of the pixels. k_u and k_v are the scale factors in the u and v direction respectively. Finally, the origin of the image plane is translated so that the center of the image plane lies on the y axis.

$$u = u_0 + \frac{k_u x' f}{y'} \tag{3.21}$$

$$v = v_0 + \frac{k_v z' f}{y'} \tag{3.22}$$

Placing the above equations in matrix form gives equation 3.23.

$$I = \begin{bmatrix} k_u f & u_0 & 0 & 0 \\ 0 & v_0 & k_v f & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{3.23}$$

Combining I and W gives the camera calibration matrix C.

$$C = IW \tag{3.24}$$

There are 11 equations given by the calibration matrix C with 16 unknowns. Note that nine of the unknowns can be specified in terms of the three unknowns θ, ϕ, ψ. This gives only 10 unknowns, but also a set of nonlinear equations.

Ganapathy leaves the equation in a linear form and applies the constraint that R is a proper orthonormal matrix. R is the 3x3 rotation matrix in W. Since R is a proper orthonormal matrix, $R^{-1} = R^T$. This constraint provides the five needed equations to solve a linear set of equations.

Using Ganapathy's technique, the camera calibration matrix is decomposed. The information provided by the decomposition is used to position the graphical cameras to match the real cameras.

3.4 Camera Platform Calibration

When the cameras are placed onto the camera platform their position in space is no longer fixed. As the camera platform moves the image centered coordinate system, represented by matrix W, is changing. Hence, a new matrix W' that changes with the motion of the camera platform is required to register the stereo graphics with the stereo cameras.

To calculate the matrix W', the camera platform was treated as a two degree of freedom robot. Once the cameras are calibrated, the matrix I is used to calculate the

matrix W'. This is done by placing the cameras on the camera platform. Now, each image and world point pairs provides an equation the constraints the location of the end effector of the robot.

The robot calibration provides several values, the base position of the robot, the initial orientation of the robot, and the orientation and lengths of the robot's links.

The rules for establishing how adjacent links of a robot are related is based on the *Denavit-Hartenberg* notation [1]. The Denavit-Hartenberg notation uses a 4x4 matrix representation to relate adjacent links in an open kinematic chain. Each joint of the links is represented by a coordinate frame. The ith coordinate frame is the joint$_{i+1}$ between ith and $i+1$ links. The three rules for establishing the axes for each coordinate frame are as follows:

1. The z_{i-1} axis is along the axis of motion of the ith joint.
2. The x_i axis is normal to the z_{i-1} axis, and pointing away from it.
3. The y_i axis completes the right-handed coordinate system.

The relative position and orientation of the coordinate frames for the two links describes the relationship between the links. Four parameters are needed to establish the relative location of the two coordinate frames are as follows:

a_i is the shortest length from z_{i-1} to coordinate frame$_i$.

d_i is the distance between the coordinate frame$_{i-1}$ and the point on the z_{i-1} that is closest to the frame$_i$.

α_i is the offset angle about the x_i axis needed to align the z_{i-1} axis with the z_i axis (using the right-hand rule).

θ_i is the angle about z_{i-1} axis needed to align x_{i-1} and x_i (using the right-hand rule).

Using matrices for translation and rotation (3.13-3.17), a set of nonlinear equations based one the above parameters is created. These equations allow the matrix W' to

be calculated for different orientations of the camera platform. The equations are solved using an iterative numerical method starting with estimated values for the different parameters.

The code for calibration of the camera platform was written by Collin Selleck, a member of the technical staff at SNL. Data points were collected using the same method as the camera calibration matrix, but the orientation of camera platform is varied during data collection.

3.5 Inverse Perspective

To calculate the line in world space that corresponds to a point in image space, the relationship between focal length and depth is used. This relationship is contained in the camera calibration matrix. Starting with basic geometry the steps involved to extract a line equation from the camera calibration matrix are outlined below.

The general equation for a plane in 3-space is given by equation 3.25.

$$ax + by + cz + d = 0 \tag{3.25}$$

The intersection of the two planes form a line. The direction of the line is calculated by taking the cross product of the normal vectors.

$$(\lambda, \mu, \nu) = (a_1, b_1, c1) \times (a_2, b_2, c2) \tag{3.26}$$

Given that a line must pass through a specific point and combining with the direction of a line, the general line equation can be derived.

$$\frac{x - x_0}{\lambda} = \frac{y - y_0}{\mu} = \frac{z - z_0}{\nu} \tag{3.27}$$

The two linear equations (3.10 and 3.11) used to create the least squares solution form two planes in world space. Once the element values of the matrix C have been calculated, they can be used to find x, y, and z values given an image point (U,V). From these two equations the value of normal vectors would be as follows:

$$a_1 = C_{11} - C_{31}U \quad b_1 = C_{12} - C_{32}U$$
$$c_1 = C_{13} - C_{33}U \quad d_1 = C_{14} - U$$
$$a_2 = C_{21} - C_{31}V \quad b_2 = C_{22} - C_{32}V$$
$$c_2 = C_{23} - C_{33}V \quad d_2 = C_{24} - V$$

Using the line equation, the normal vectors from the two planes, and a specific z location (z_0), the corresponding x and y location can be calculated. The corresponding x_0, y_0 values are computed with equations 3.28 and 3.29.

$$x_0 = \frac{b_1(c_2 z_0 + d_2) - b_2(c_1 z_0 + d_1)}{a_1 b_2 - b_1 a_2} \tag{3.28}$$

$$y_0 = \frac{a_2(c_1 z_0 + d_1) - a_1(c_2 z_0 + d_2)}{a_1 b_2 - b_1 a_2} \tag{3.29}$$

3.6 Algorithm

The algorithm implemented for the computer vision system to obtain dimensional information uses a feature extraction algorithm and an area based matcher.

Area based matching algorithms assume that each pixel or point in the left image has a corresponding pixel in the right image. To find the corresponding pixel in the right image a correlation process is carried out. First a window is drawn about the pixel in the left image. This window is then correlated with all possible matching windows in the right image. The window that has the highest correlation value provides the matching pixel. The correlation process used is based only on pixel intensities.

Occlusion presents the biggest problem to the area matching scheme. Occlusions occur when a portion of one image does not appear in the other. Therefore, a pixel in the left image would not have a matching pixel is the right image. This problem will lead to false matches with an area based matching scheme.

For the prototype computer vision system it is going to be assumed that occlusions do not exist. In other words, the objects of interest will be in full view by both

cameras. This assumption is valid in several cases because the position of the camera platform is controlled by the operator of the IGMS.

Equation 3.30 is used to find the matching pixel. When the squared error is minimized a match is found. This algorithm was selected over the standard correlation because the match or correlation is taking place at an edge boundary. A typical edge boundary will have both very high and very low pixel intensities. Therefore, if a correlation were carried out normalization would be required.

Expanding equation 3.30 gives equation 3.31. By examination, it is seen that equation 3.31 is related to the correlation process. The term $\sum\sum fg$ is the standard correlation between f and g. Taking the derivative of 3.30 with respect to g, it can be shown that the error term is minimized when $f = g$. This corresponds to the normalized correlation of f and g, as shown in equation 3.32, being maximized (equal to 1).

$$E = \sum\sum(f-g)^2 \tag{3.30}$$

$$E = \sum\sum f^2 - 2\sum\sum fg + \sum\sum g^2 \tag{3.31}$$

$$C_{norm} = \frac{\sum\sum fg}{\sqrt{\sum\sum g^2}\sqrt{\sum\sum f^2}} \tag{3.32}$$

Matching algorithms are usually restricted to matching image points that lie on a epipolar line in the other image plane. An epipolar line is the intersection of the plane formed by an object or image point and the points of focus of the two cameras with the two image planes. Since a rough estimate of the object depth is known, the search can be further restricted by taking a line segment that corresponds to the image point to be matched and the depth information provided by the operator. The depth information provided by the operator is assumed to have a $\pm$ three inch error. This error range is larger than that shown by testing with the aid of the ultrasound.

Feature extraction algorithms provide symbolic descriptors of an image. The feature that is needed for the IGMS is the boundary between different objects. The

IGMS approximates this by locating edges. An edge is the boundary between two regions with different gray-levels. Edge detection locates these discontinuities in an image. An edge detection algorithm developed by John Canny [5] is used to extract this feature form the image.

To extract edge elements (edgels) the image is convolved with the first directional derivative of a Gaussian smoothed in both the u (horizontal) and the v (vertical) direction. The technique which Canny refers to as non-maximum suppression is used to locate peaks in the gradient image. The resolution of the edges is controlled by varying the standard deviation of the Gaussian filter. Equations 3.33 and 3.34 show the relationship of the Gaussian filter with respect to the the filter width w.

$$\left(\frac{-x}{\sigma^2}\right) e^{-\frac{x^2}{2\sigma^2}} \tag{3.33}$$

$$\sigma = 2\sqrt{2}w \tag{3.34}$$

Since the edgels are used to define the boundary of the object to be modelled, a small width is desired. Therefore, w is set equal to 3. The assumption is made that the gross feature extraction has been carried out by the human operator. As the width of the gaussian filter is narrowed more detail, but also more noise will enter into the edge extraction process.

To control the noise problem several steps were taken. First, the objects to be modelled were white against a black background. Second, the lighting of the test environment is controlled. Third, the threshold for edge acceptance is set higher than would otherwise be required.

The edgels in the left image is not matched against edges in the right image, but are only used to identify object boundaries. In other words, the matching process explained relies on segmentation provided by the user not the edgels extracted from the images.

3.7 Triangulation

Having one camera allows 3D points to be mapped to a 2D points, but the mapping process is not reversible. In other words, spatial information is lost in the mapping process. One method to recover the lost spatial information is triangulation. By introducing a second camera, the mapping from two 2D image points to a 3D world point allows spatial information to be recovered.

After two image points are matched, the location of the object point is calculated using triangulation. Each image point provides a ray into world space which is calculated using equation 3.27. The rays intersect at the location of the object point. To understand the accuracy of the IGMS, the factors that affect triangulation are outlined below. The ray projected from the camera center has some associated angular error. This error is the sum of several errors: calibration error, marking and matching error, and error due to pixel resolution. Pixel resolution is increased by decreasing the camera's field of view. Marking error is based on the edge location algorithm used. Matching error is determined by the area based matching algorithm implemented. Error introduced by the camera system during calibration can be improved by collecting more data points or using a more complex model for both the cameras and camera platform. It should be noted the triangulation error scales with the distance of the object from the cameras.

3.8 Images to Objects

The basic concept in moving from images to objects with the IGMS is to have the operator do the high level image processing and allow the computer vision system to do the low level image processing. The operator segments the scene using the graphical pointer and identifies the model to be created. The graphical pointer provides 3D points which can be converted into two image points (left and right) using the camera calibration matrices. These points are used as a first cut at segmenting the scene. They are also used to restrict the size of the 3D model being created. The

IGMS has two models, cylinder and box, that the operator can select from for the purpose of identification.

The computer vision system does two forms of low level processing on the image. One is locating edges which are used to identify the boundary of the object to be modelled. The other is finding corresponding pixels in the left and right image. Once this is completed, key features on the object's boundary are used to calculate position and dimensions. This information is processed by the CAD package to create a graphical object.

Below is a description of how the edgels are processed to create the graphical objects.

3.8.1 Cylinder

The first object recognized by the IGMS was a cylinder. The side view of a cylinder provides all the information necessary to construct a graphical three dimensional cylinder. The image points provided by the graphical pointer bound the object. These points are stored in array_0. The first step is to find the boundary or edge of the cylinder in the left image. The search for the edge of the cylinder boundary starts by the selecting two pixels in array_0 with the lowest u value. The estimated edge position is the average of the u and v values of these two pixels. From this image point the search is expanded in all directions until an edge is located. If the expansion is more that 15 pixels in any direction the search will stop and no cylinder will be located.

Once a valid edgel (e_0) is located it is stored in array_1. The edge is followed by performing a search on all adjacent pixels. To accomplish this all edgels that neighbor e_0 are placed onto stack_0. The top pixel on stack_0 is popped from the stack and placed into array_1. It's neighboring pixels which are edges and are not in array_1 are pushed onto stack_0. This process continues until stack_0 is empty. Array_0 now holds all the connected bounding pixels. The number of pixels in the array is compared with the estimated number pixels of that should bound the cylinder. The estimate is calculated from the initial bounding pixel values. If these values do not agree, the search begins

for a new set of bounding edgels. Figure 3-1 shows an example of the bounding pixels that are located.

The key feature for the cylinder is the corners. The corners are located by examining the phase change from the neighboring pixels. If the phase change is greater than 25 degrees a possible pixel corner has been located. Each possible corner is placed into $array_2$ The four corner pixels for the cylinder are selected from the possible corners by comparing their distance to the bounding points. The closest pixel in $array_2$ to each bounding pixel is selected as the corner pixel. The corresponding corner pixels in the right image are located using the area based matcher. From these corner pixels in the left and right image 3D world points can be calculated. Given the assumption that the height is larger than the radius, these points are used to calculate height, radius, and position of the cylinder.

One drawback to this method is that a top view of the cylinder will not return the proper dimension or position. Since the operator controls the camera position this drawback was ignored.

This method for extracting geometric information about a cylinder is extremely prone to error if false edges exist or if edges of objects in the background are followed. Since the matching algorithm limits the depth of the search, background edges will be rejected by system when the constraint is added that each edgel must have a corresponding image point. In other words, when the matching error exceeds a fixed threshold the matching point is thrown out.

3.8.2 Block

To increase the flexibility of the IGMS, bounding boxes were added. The box is created by extracting the maximum and minimum value for each axis. Each edgel in the area marked by the graphical pointer in the left image is matched to the right image. The match, as before, is only accepted if it is below a specified threshold. The 3D points are triangulated from corresponding image points. The 3D points are sorted for maximum and minimum values on each axis. This provides dimension and position for the block.

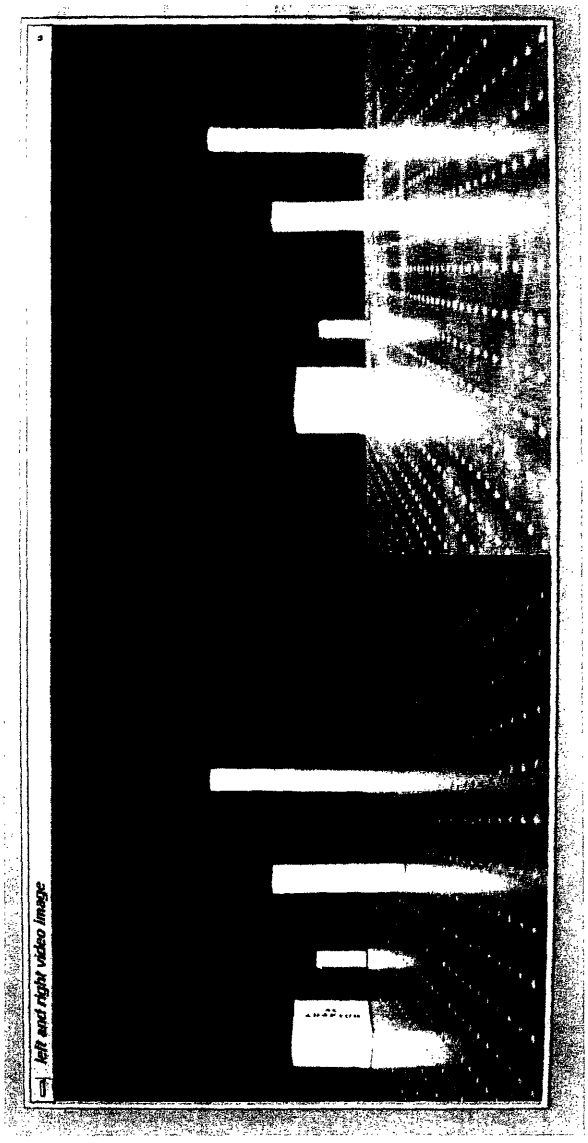

Figure 3-1: Bounding Pixels of a Cylinder

Due to the crudity of this method, texture mapping is applied so that the real object may be viewed in the graphical workcell. The coordinates for the texture map are provided by the maximum and minimum u,v values. The information provided by the texture map improves object identification.

Chapter 4

Testing of IGMS

The IGMS was constructed in stages so that a step by step debugging and testing would be possible. First, the cameras were tested for matching and correspondence. Next, the user's ability to position the graphical pointer was tested. The camera platform was tested for angular accuracy. Finally, the process of converting image to objects was tested. A description of the different tests is given below.

4.1 Testbed

The testbed for the IGMS was constructed in the VR Laboratory at SNL. The floor of the testbed is a six feet by four feet optical table. The table made measurement of an object's position simple and accurate. Cylinders used for testing varied in both height and radius. The heights range from 3 inches to 12 inches. The radius is either 0.5 or 0.25 inches. The dimensions of bounded objects varied in a similar fashion. The cameras are positioned approximately 24 inches from the end and 6 inches above the optical table.

4.2 Camera Matching

One of the keys to setting up the IGMS is to match the left camera to the right camera. Matching the cameras allows the operator to fuse the two 2D images into

one 3D image. To match the cameras for fusability the orientation of the cameras must be aligned and the field of view and aspect ratio must be matched as closely as possible.

The optical axes of the two cameras are aligned by adjusting the angles (pan and tilt) of the cameras. The pan angle is adjusted until corresponding vertical lines in the two images are parallel. The vertical lines are provided by the optical table. The alignment process is manually done using a monitor. Both images are displayed on the monitor one in blue the other in red, this makes it easy to tell when the vertical lines are parallel.

The tilt of the cameras is aligned by placing both cameras on a flat platform. This alignment method does not allow for exact horizontal alignment, but is close enough for the operator to fuse the two image.

Field of view is adjusted in both camera until an object placed an equal distance from both camera fills the same number of pixels horizontally and vertically in the two images.

The accuracy of these adjustments can be measured once the cameras are calibrated. Shown below are the camera calibration matrices for the left and right camera after the above adjustments have been carried out. These matrices are decomposed to examine the difference in the two cameras after alignment.

$$C_{left} = \begin{bmatrix} -232.71 & -64.35 & 4.45 & 5031.32 \\ -5.22 & -63.50 & -285.65 & 1379.56 \\ -0.01 & -0.24 & 0.01 & 1.00 \end{bmatrix} \tag{4.1}$$

$$k_u = -944.89 \qquad k_v = 1191.53$$

$$u_0 = 323.6 \qquad v_o = 213.0$$

$$x = 20.85, y = 3.06, z = 3.77$$

$$\theta = 1.51, \phi = 0.043, \psi = 0.00$$

$$C_{right} = \begin{bmatrix} -214.44 & -60.906 & 4.84 & 3844.42 \\ -4.93 & -58.97 & -261.44 & 1294.73 \\ -0.01 & -0.22 & 0.01 & 1.00 \end{bmatrix} \tag{4.2}$$

$$k_u = -945.29 \qquad k_v = 1175.58$$

$$u_0 = 309.1 \qquad v_0 = 217.7$$

$$x = 16.89, y = 3.96, z = 3.74$$

$$\theta = 1.53, \phi = 0.039, \psi = 0.01$$

The least squared error values for k_u and k_v show the accuracy achieved by the alignment process for pixel size. The k_u had a 0.1 percent error and k_v had a 1.3 percent error.

The angular alignment error can be calculated for the orientation angles θ, ψ, ϕ. θ, ψ, ϕ had errors of 1.2, 0.19, 0.29 degrees respectively. The angular error in θ is offset by the mismatch of the image centers. Otherwise, the angular error in θ would not be a close match between the two cameras.

After matching the real cameras, the graphical cameras are created in *CimStation*^{TM}. The real cameras are fixed in position to isolate the camera model from the camera platform model. To test the accuracy of the graphical overlay, graphical models of the real cylinders are manually entered into *CimStation*^{TM}. The real cylinders are measured and placed onto the optical table. This provides the pose and dimension entered into *CimStation*^{TM}. Using a flat screen monitor, the accuracy of the graphical overlay is examined. It was shown that the graphical overlay has an error of less that five pixels in both the u and v direction for a cylinder with a height of 3 inches and a radius of 0.5 inches. This measurement was made by shifting the images with respect to the wireframe overlay. Pixel error appeared to scale with size of objects.

The more important test was the appearance of the overlay in the stereoscopic viewer. The graphical cylinders that were manually entered visually corresponded in

the real cylinders. To further test the accuracy, the positions of the cylinders were incorrectly entered into *CimStation*^{TM}. In the stereoscopic viewer horizontal offset was detected when the object was moved 0.25 inches or more. When the depth of the cylinder was changed detection was poorer. The cylinder could be moved up to two inches before it was detected. However, on the flat screen monitor these offsets could be detected sooner.

One important factor when setting up the cameras is camera separation. As the separation of the cameras increases fewer people are capable of fusing the two 2D images in the stereoscopic viewer. It was determined that a camera separation of greater than 4 inches presented a problem for most users with the camera field of view approximately 25 degrees and an object distance of 3 to 7 feet from the cameras.

4.3 Graphical Pointer

The next step tested motion control of the graphical pointer. The first attempt used a magnetic tracker to position the pointer based on the location of the operator's wrist. This attempt failed due to noise in the tracking system and the difficulty in placement of the tracked device. The tracking system was replaced with a joystick. The joystick is less intuitive than the tracking system for motion control. Using the tracking system to move the pointer upward the user raised his hand, to move the pointer upward with the joystick it is twisted counter clockwise. The joystick while confusing for some users did perform adequately.

The placement of the graphical pointer was tested by having the operator position the pointer over real objects in the scene. Placement over virtual objects was not tested or compared. The testing suggested that there are two classes of users. One class could position the pointer within 2 inches of the objects depth. The other class could not position the depth of pointer with any accuracy. Both classes could position the pointer to 0.1 inch horizontally and vertically once the pointer was set at the correct depth.

These test results were the reason that the ultrasonic ranging was added to the

system. With the depth ranging capability, any user could correctly position the graphical pointer to segment the scene for the computer vision system.

4.4 Camera Platform

The camera platform calibration was tested. During the calibration process inaccuracies in the camera platform were detected. When the camera platform was tilted from 0 degrees to 20 degrees and back the camera platform would not return to the starting position. This was detected by placing a ruler so that it filled the entire vertical field of view and examining the value that the camera platform returned to. The value varied by an inch at a distance of 3 ft.. This corresponds to an angular error of 1.59 degrees. With this type of error registration between the graphical environment and real environment is difficult.

The following is a calculation showing the expected error in the camera platform. The theoretical accuracy of the camera platform is based upon the type of encoders used and the gear ratio between the encoders and the camera platform. Below is a calculation for the expected angular error for a tick of the encoder. A tick being the smallest controllable unit for the camera platform. Two different encoders were used to control camera platform. The yaw encoder has $\frac{1800ticks}{quadrant}$. The gear ratio for the rotation about the z axis is 13.57. The pitch and roll encoders have $\frac{1000ticks}{quadrant}$. The gear ratio for rotation about the x and y axes is 5.14.

Yaw:

$$\frac{1}{\frac{4\times1800}{360}\times 13.57} = \frac{3.684\times10^{-3}degrees}{tick}$$

Pitch or Roll:

$$\frac{1}{\frac{4\times1000}{360}\times 5.14} = \frac{1.750\times10^{-2}degrees}{tick}$$

To evaluate the above values they are compared with approximate pixel resolution.

Pixel resolution:

$$\frac{23degrees}{480pixels} = \frac{4.792 \times 10^{-2}degrees}{pixel}$$

Using these numbers, the camera platform should produce better than pixel level accuracy in all three degrees of freedom. But due to play in the gears, the camera platform was only able to produce pixel level accuracies for rotation about the z axis.

Initially, the camera platform was going to track the viewer in all three degrees of rotation, but due to inaccuracies in roll and tilt motion it was only tracked about the z axis. In other words, the operator could only pan the camera platform. With this restriction the camera calibration was carried out. Using the stereoscopic viewer, registration of the graphical cameras with the camera platform was tested. The flatscreen display was used to search for the error. The error associated with the camera platform model was visually undetectable.

4.5 Computer Vision

To determine the window size for the matching algorithm, the computer vision software was tested. The tested window sizes for the area based matcher were localized. They varied in size from a 3x3 to a 13x13 window. Pixels in the left image that lay on an edge that was vertical to the epipolar lines were easily matched with all window sizes. The problem for the matching algorithm was caused by pixels that laid on edges that were approximately parallel to the epipolar lines.

Due to the fact that the camera model provides an estimate of the epipolar lines, the highly localized 3x3 matching window performed poorly. It has a pixel offset of three or more when locating the corresponding pixel in the right image. This performance could have been improved by requiring connectivity of 3D points. To accomplish this all bounding pixels would have had to been matched, instead of the four key corner pixels of the cylinder.

The 10x10 window performed best with a maximum of one pixel offset from the corresponding pixel. The 10x10 would not perform well if the background varied greatly, but for the test setup the background was relatively uniform.

4.6 IGMS

The IGMS was tested by several inexperienced operators during different stages of development. Most of them were able to build graphical models using the system after a short explanation about the system.

From experimental results, the graphical cylinders created by the IGMS system were within 0.1 inches of actual dimension. The position or depth accuracy was with 1.5 inches of measured distance. This is due to the fact that the camera separation is small (4 inches). These results show that this type of system would be adequate for positioning robot manipulators and tools.

Chapter 5

Conclusions

The IGMS demonstrated a method for gathering geometric data from a remote site. The geometric data collected is displayed in a graphical environment which is overlayed onto video of the remote site. This allows visual verification of graphical models. The key feature that makes visual verification and data collection possible was the use of virtual reality. The virtual or graphical environment was used for the verification process and also assisted in the model building process through the use of a 3D graphical pointer.

A step by step method for registering the graphical environment was outlined. This method allows the user to fuse both the real and graphical 2D images into one 3D image. The calculations required to register the graphical environment, also allows the computer vision system to make accurate measurements in the real environment.

The graphical pointer proved useful for rough segmentation of the scene. It could also be used to locate key features. The graphical pointer, unlike a sensor array, could be moved about the remote environment without fear of destroying equipment.

The sound feedback and voice recognition system added an intuitive interface to the system. Voice commands meant that the user no longer needed to memorize which buttons mapped to a specific function. The sound feedback kept the user informed on the state of the system.

Finally, the geometric data collected was transformed from raw data into objects by having the operator do the high level recognition of the objects. This concept

proved useful because specific code could be written to create different types of objects.

The IGMS provides another block in the chain for research on geometric data collection from a remote site.

5.1 Future Work

The IGMS demonstrated several important points, but several limitations of the system are obvious. Below is a description of possible improvements.

A camera platform with a higher level of accuracy in three degrees of freedom (pan, tilt, and roll) is needed. This would give the operator a greater feel of immersion since more of the head motion could be tracked. More of the remote environment can be examined with increase freedom of motion.

The accuracy of the models created needs to be improved. This includes both a better method for collection of raw geometric data and method for finding the key features associated with each object. One method for improving data collection would be to use multiple cameras for better triangulation. Another method would be a sensor fusion approach using a laser range finder in conjunction with the stereo video.

A more robust approach to object recognition should be incorporated into the system. This approach should allow several different views to be combined to created a single object. In other words, the system should be able to store partial information for later use in model building.

The number of objects or primitives that the operator can specify should be increased. This would require finding key features of each new primitive.

The IGMS user interface still has room for improvement. More complete instructions should be included with the sound feedback. The instructions should be setup so that no training is necessary to use the system.

In conclusion, the IGMS has demonstrated some of the benefits of combining a graphical environment with a real environment, but more work is still required before

the benefits can be fully obtained.

Bibliography

[1] Asada, H, and J.E. Slotine, *Robot Analysis and Control*, Wiley, New York, New York, 1986.

[2] Ballard, D. H. and C. M. Brown, *Computer Vision*, Prentice-Hall, Englewood Cliffs, New Jesrey, 1982.

[3] Bolas, M., and Mcdowall, I., *Boom User's Guide*, Fakespace, Menlo Park, California, 1992. Refrence: Part Number 2C-9001

[4] Bon, B., T. Litwin, and D. B. Gennery (1990) "Operator-Coached Machine Vision for Space Telerobotics," *Proceedings of the SPIE - Cooperative Intelligent Robotics in Space*, vol. 1387, pp. 337-342.

[5] Canny, J. F., (1986) "A Computational Approach to Edge Detection, " *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8 no.6, pp 679-698.

[6] Christensen, B. K., Drotning, W. D., and Thunborg, S. (1992) "Model Based, Sensor Directed Remediation of Underground Storage Tanks", *Journal of Robotic Systems*, pp. 145-159.

[7] Ganapathy, S. (1984) "Decomposition of Transformation Matrices for Robot Vision" *IEEE Robotics* pg.130-139.

[8] Green, J. *FlyBox Owner's Guide*, BG Systems, Palo Alto, California, 1992.

[9] Gonzalez, R.C. and P. Wintz, *Digital Image Processing,* Addison-Wesley, Reading, Massachusetts, 1987;

[10] Horn, B. K. P., *Robot Vision*, Mit Press, Cambridge, Massachusetts, 1986.

[11] Milgram, P., D. Drascic, and J. Grodski (1990) "A Virtual Stereographic Pointer for a Real Three Dimensional Video World," *Proceedings of the IFIP TC - Human-Computer Interaction. INTERACT '90* Elservier Science Publishers: Amsterdam. pp. 695-700.

[12] Oxenberg, S. C., Landell B. P., and Kan, E. (1988) "Geometric database maintenance using CCTV cameras and overlay graphics," *Proceeding of the SPIE - Space Station Automation IV*, vol. 1006, pp. 115-123.

[13] Schalkoff, R. J. *Digital Image Processing and Computer Vision*, Wiley, New York, 1989.

[14] Strang, G., *Introduction to Applied Mathematics*, Wellesey-Cambridge Press, Cambridge, Massachusetts, 1986.