MULTIPROCESSOR STEREOSCOPIC VISUALIZATION SYSTEM FOR IMAGES, VIDEO AND HIGH DIMENSIONAL DATA

By

Roger Duane King
A Dissertation
Submitted to the
Graduate Faculty
of
George Mason University
in Partial Fulfillment of
The Requirements for the Degree
of
Doctor of Philosophy
Computational Sciences and Informatics

Committee:

Edward J. Wegman — Dr. Edward J. Wegman, Dissertation Director

Dr. Jim Chen, Committee Member

Dr. Ruixin Yang, Committee Member

Stanley M. Zoltek — Dr. Stanley M. Zoltek, Committee Member

Dr. Michael Summers, Director, School of Physics, Astronomy, and Computational Sciences

Dr. Timothy L. Born, Associate Dean for Student and Academic Affairs, College of Science

Dr. Vikas Chandhoke, Dean, College of Science

Date: APRIL 27, 2012

Spring Semester 2012
George Mason University
Fairfax, VA

Multiprocessor Stereoscopic Visualization System for Images, Video and High Dimensional Data

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at George Mason University

by

Roger Duane King
Master of Science
George Mason University, 1992
Bachelor of Arts
George Mason University, 1986

Director: Edward J. Wegman, Professor
School of Physics, Astronomy, and Computational Sciences

Spring Semester 2012
George Mason University
Fairfax, VA

DEDICATION

To twin shooting stars whose glow filled our lives for too short a time

Veronica Lee Maddox

This Dissertation is dedicated in loving memory of Veronica Lee Maddox and Dolores Jean Maddox-Martin. Vicki and Lori were identical twins born on December 12, 1968. They were both star athletes in High School, excelling in tennis, basketball and soccer. Vicki passed away May 27, 2011 and her twin sister Lori passed away on November 22, 2011. They were both stunningly beautiful girls. Vicki was kind and compassionate; she always liked animals, I still remember her telling me about feeding deer from her hand while she was in the mountains with her parents. She was charming and was the kind of girl that could make a sweatshirt be an elegant ball gown. She truly loved her son Joey; whenever we talked, Joey was sure to come up. Lori seemed to have endless energy, she loved planning the next dish she would cook for her two boys, Chris and Matt. They came from a large family and the loss of the girls has been devastating for their families and friends. The day doesn't pass without me thinking of them and missing them.

Dolores Jean Maddox-Martin

ACKNOWLEDGEMENTS

I would like to thank my committee members, Dr. Jim Chen, Dr. Ruixin Yang, and Dr. Stanley Zoltek for their time and guidance during the dissertation process. My adviser, Dr. Ed Wegman has been a true mentor during my academic career. I don't think I could have made it this far without his help and guidance. I will always be proud to be known as one of his students.

I would also like to thank the Academic Affairs office of the College of Science. Melissa Hayes, Emmy Daniels, Laura Kasik, Bertha McCoy, Gail Hodges, and Jennifer Bazaz, I appreciate all your help.

Finally, I would like to thank my family and friends. My parents have always been supportive of my academic endeavors, and I would not be here without their help, support and love. My dear friend, Tamara Maddox (without whom I might never have receive my undergraduate degree), her husband John, and their children, Joshua, Danny, and Tammy, have all provided wonderful support and kind encouragement. Most particularly, I want to thank Joshua for inspiring me to complete this challenging process. Joshua has been starring as "Jojo, the Mayor's son" in Seussical at the Riverside Dinner Theater for the last few weeks. At a time when finishing this degree seemed overwhelming (and perhaps impossible), it has been truly inspiring hearing him sing "Anything's Possible" on stage, week after week.

Thanks to all of you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

ABSTRACT

MULTIPROCESSOR STEREOSCOPIC VISUALIZATION SYSTEM FOR IMAGES, VIDEO AND HIGH DIMENSIONAL DATA

Roger Duane King, Ph.D.

George Mason University, 2012

Dissertation Director: Dr. Edward J. Wegman

Humans visualize information in three dimensions and use a stereoscopic system for depth cues. This visualization system uses LCD projectors to create a three dimensional data visualization system. Each of two projectors use a polarizing filter to isolate its image to one eye or the other of a human viewer wearing simple glasses with corresponding polarizing filters. The two projectors are aligned and display an image in the same space on a lenticular screen. Each projector is driven by a computer. The display computers are controlled by a separate system which provides for all user interaction. This visualization system is able to present three dimensional images stored in a number of formats to the user. The visualization system is also able to display multi-dimensional data to the user in parallel coordinates or within a Cartesian Coordinates framework.

CHAPTER ONE

Background, Previous Work, and Focus of this Dissertation

In the context of this dissertation, I will take 3D and stereoscopic to mean the same thing, i.e. that there are means for presenting slightly different images to the left eye and to the right eye so that the human visual system can integrate them into a single image that is stereoscopic. Sometimes, 3D is taken to mean an image that obeys the principles of perspective with additional lighting and rendering models so that depth cues can be inferred from the image without a true stereoscopic display.

The history of 3D stereoscopic visualization can roughly be divided into three eras. From 1838 to about 1930, devices such as the stereopticon coupled with the newly developing art of photography made 3D stereoscopic a fashionable technology and popular especially among the people of the Victoria era. From about 1950 to perhaps the early 1970s, 3D stereo gained new interest with the more extensive use of anaglyph stereo on the printed page and polarized light stereo in the world of motion pictures. Beginning around 1990 with the introduction of high refresh rate CRT projectors and liquid crystal shutter glass until the current time with 3D high definition television sets, stereo has gained additional interest among scientists as well as movie makers and the general public.

Stereo Visualization Technologies and Devices

Symanzik (2011)[1] outlines a number of stereoscopic display systems:

- Time-Parallel
  - Anaglyph
    - Red-Green
    - Red-Cyan
  - Polarized Light
  - Separate Image
    - Split Screen
    - Dual Screen
    - Head-Mounted
  - Freeviewing
  - Autostereogram
- Time-Multiplex
  - Electro-optical
    - Liquid Crystal
    - Shutter Glasses.

He also mentions several other techniques which have not gained any substantial following. The stereopticon in the earliest era is an example of the Separate Image Viewer. Anaglyph and Polarized Light Stereo are technologies most represented in the middle era. Shutter Glasses, usually built around liquid crystal devices, are the popular choices in the present era of stereo display devices.

The underlying principles of stereoscopic (binary) vision was documented by Wheatstone[2] but he makes suggestions that some of the geometric diagrams of Euclid were intended as stereoscopic devices. He also suggests that Leonardo da Vinci (1452 – 1519) and Gaspar Monge (1746 – 1818) knew of and made contributions to the understanding of stereoscopic binary vision. Wheatstone's 1838 paper, however, describes a (complex and bulky) device for stereoscopic visualization and as such must be considered the pioneer in stereoscopic visualization. (It should be noted that Charles Wheatstone is often given credit for the invention of the electrical circuit called the Wheatstone Bridge, although it was actually invented by Samuel Hunter Christie in 1833, and later improved by Wheatstone in 1843.) Although the principles of stereoscopic vision were understood, it was not until photography emerged that a viable method of production of stereo images became available. By 1850 a variety of photographic processes including tintypes, albumin, and daguerreotypes were developed.

> "Photographers around the world produced millions of stereoscopic views between 1850 and 1930. Their popularity soared when Queen Victoria and Prince Albert received the gift of a stereoscopic viewer at the Crystal Palace exhibition in 1851. Soon after, the American jurist Oliver Wendell Holmes called for the establishment of 'special stereographic collections just as we have professional and other libraries.' Around the world, independent and entrepreneurial photographers broke into the growing market for illustrations of all types of subjects: local history and events, grand landscapes, foreign monuments, charming genre scenes, portraits of notables and urban architecture. War and disasters such as floods, fires, train-wrecks, and earthquakes were enormously popular subjects." New York Public Library (2011)[3].

Figure 1 Stereopticon circa 1890. From E. Wegman Collection

Oliver Wendell Holmes (1809-1894) together with businessman Joseph L. Bates (1807-1886) developed a popular hand-held stereo viewer in 1859. From that time until the 1930s stereoscopic viewing of side-by-side images became a major recreational activity. It should be noted that broadcast radio became accessible in the early 1920's so that radio tended to supplant stereopticons as a technological marvel. See Phillips (2006)[4] for a more elaborate discussion of the *first era* of stereoscopic display devices.

The earliest 3D movies were actually filmed around 1915 using anaglyph glasses, basically as demonstration shorts. Of course, only black and white film would have been available then. Frequently early silent movies would be tinted to set the mood, so some coloring would have been possible. These were followed by a series of short films, which primarily lasted until the period 1952 to 1954, when a number of feature length films were made using polarized filter glasses. Best known among the era of 3D movies were:

- Bwana Devil – 1952
- Kiss Me Kate – 1953
- House of Wax – 1953
- Hondo – 1953
- Dial M for Murder – 1954
- Creature from the Black Lagoon – 1954

Woods (2012)[5] identifies approximately 540 3D films that were theatrically released with some 206 of those released since 2007 and some 88 3D films including shorts released between 1952 with Bwana Devil up to 1956. The 1980s saw some resurgence in 3D movies although these tended to be B movies, some on the seamier side of movie fare. Of course those films released in the 1950s and through 1980s generally exploited polarized light filters, while releases on VHS tapes exploited anaglyph technology. The film releases in the 2000s generally used shutter glasses. Those released for home use Blu-Ray discs and stereo capable HDTV sets. Anaglyph 3D comic books (graphic novels) were also popular in the 1950s concurrent with the 3D movie craze. These were presented in anaglyph form usually with red-cyan glasses. Red and cyan are complimentary colors and in an additive color system, they add to white. In contrast, red and green add to yellow in an additive color system, hence give an overall yellowish cast to the 3D image[6]. Examples of the 1950s era 3D comic books are hard to come by because they are highly collectible and consequently expensive.

Symanzik (2011)[1] recounts the early history of anaglyph stereo techniques and attributes the earliest work to Rollman (1853a,b) and d'Almeda (1858)[7]. According to Symanzik, anaglyph displays have been used beginning in the 1960s in chemistry, cartography, biology, and medicine. Applications of anaglyphs to mathematics, statistical methods, and data visualization lagged other disciplines somewhat. Between 1983 and 1986, Daniel Carr and his colleagues described the use of anaglyph stereo in statistical graphics. Probably an early landmark book containing anaglyph stereo images by Carr, of Tom Banchoff, and the late Ruben Gabriel was Wegman and DePriest[8]. More recent work containing anaglyph statistical graphics includes Wegman and Luo[9] and Symanzik[1].

Interest in Virtual Reality in the early 1990s revived extensive interest in the use of stereo visualization for scientific purposes. The invention of CrystalEyes liquid crystal shutter glasses around 1990 and high refresh rate CRT projectors allowed for the emergence of highly satisfactory stereo visualization. While ordinary motion pictures have a frame rate of 24 frames per second and 1990's era NTSC TV had a frame rate of 30 frames per second, high end CRT projectors of the 1992 era were capable of 120 frames per second. The CrystalEyes shutter glasses were synchronized using infrared signaling to a computer output at the rate of 60 frames per second for each eye, yielding a highly satisfactory viewing experience. This technology coupled with RISC based, graphics-oriented Silicon Graphics Machines provided ideal, if expensive, platforms for experimentation with genuine 3D data visualization frameworks. Cruz-Neira et al[10] reported on the development of the CAVE. (Cave Automatic Virtual Environment). Cruz-Neira upon graduation took a faculty position at Iowa State University where she collaborated with Dianne Cook and others to experiment with the development of statistical graphics applications. Approximately the same time by 1992, independently of the Cruz-Neira efforts, Dr. Wegman and his colleagues installed a similar projection system used for data visualization and statistical graphics also using CrystalEyes shutter glasses, Silicon Graphics computers, and a high refresh-rate CRT projector. A series of data visualization software programs were developed including Mason Hypergraphics (1988, 1989), Mason Ridge (1990, 1991), ExplorN (1992), and 3D MRI (1993). The experiences of Wegman and his colleagues with their immersive system were reported in Wegman and Symanzik [11]. CAVE environments require very large spaces and are quite expensive, easily $1,000,000 or more. The one-wall projection system that Dr. Wegman had developed was more economical in space (about 400 square feet) and considerably less costly, on the order of $250,000. In 1996, Dr. Wegman ran MATLAB test code on both his Silicon Graphics RISC workstation and a newly acquired 400 megahertz Pentium PC and discovered that they tied speed wise in a virtual dead heat. This set the stage for later VR developments.

Figure 2 The CAVE Environment

Focus of this Dissertation

It should be noted that the development of the technology, e.g. stereopticons, anaglyph printed documents, stereo motion pictures, CAVE environments, shutter glasses, etc. were developed largely independently of the ‘software’ that was to exploit them. Indeed, the ‘software’ has not been easily transferable from one system to another. It was not until the 1990s that computer technology could be mated to stereoscopic technology for purposes of scientific and data visualization. The goal of this dissertation is the description of the development of an integrated hardware-software visualization system consisting of a multiple computers with modules that include:

- a communication/control system
- display systems
- an image viewer system and
- a data visualization system.

These will be described in detail in the remainder of the dissertation. But briefly, the communication/control system includes a master computer which controls two or more display computers. The display computers can be configured for stereoscopic display, which I focus on primarily in this dissertation. However, the display computers can also be configured as a “wall system”. The image viewer displays images in a bit map format and can process stereoscopic image from a variety of formats including side-by-side and anaglyph as well as proprietary formats such as NASA and visible human. The data visualization system captures much of the capability of the stereoscopic software developed for the SGI/shutter glasses configuration reported on in Wegman and Luo [9]and Wegman and Symanzik[11]. The cost of the integrated system with present day hardware is on the order $5000 compared with the $250,000 for the SGI/shutter glasses system. The modules I report on encompass some 10,000 lines of code.

CHAPTER TWO

Overview

This dissertation describes a hardware and software system to render high dimensional data and stereo photo imagery to the user. The hardware of the system consists of at least three networked computers each connected to a display device and one of three has a mechanism to interact with the user. This visualization system displays stereo imagery by using two computers. One of the computers will display the image to the right eye and the other computer displays the image to the left eye. The images are displayed using projectors. The projectors are stacked one on top of the other and have a linear polarizing filter on the front of the lens. The projector assembly is seen in Figure 3. The two projectors are manually aligned to present the greatest overlapping image. Each projector has a vertical lens shift. The lower projector's image is shifted up and the upper projector's image is shifted down. There is a small amount of divergence in the two image projections but it is not distracting in practice. The images are projected onto a silver lenticular screen [12]. The silver screen is required in order to preserve polarization. The user needs to wear glasses with linear polarizing filters and the filters on the right and left eyes are out of phase. The polarizing filters on the glasses are aligned with the polarizing filters on the projectors. The polarizing glasses will allow the image from the left eye computer to be seen by the left only and the image from the right eye computer to be seen only by the right eye. This type of display is referred to as passive stereo. Another type of stereo display (which is not utilized by this system) is an active stereo display. An active stereo display system still requires the user to wear glasses. These glasses are actually LCD shutter glasses. The projection system in an active stereo display will display right and left images in an alternating fashion. Each time the projection system alternates between right and left a pulse of light is sent through an infrared transmitter. The transmitted pulse of light is received by the glasses and instructs the glasses to turn the LCD in front of one completely dark and the LCD in front of the other eye completely transparent. The glasses in an active stereo display are expensive and fragile. The projection system in an active stereo display must produce twice the desired frame rate since the user is sequentially viewing right and left eye images. Active stereo display systems are primarily used in the current generation of consumer 3D TV systems. The 3D TV is given a single input stream with the 3D frames encoded. These TVs do not have separate inputs for the right and left eye images therefore these displays are really not useable for scientific visualization. If one wished to encode a stream with the 3D information it would be necessary to acquire the proper codecs and these do not exist in the public domain. The cost for an educational institution to license the software for encoding 3D bluray dvds in March 2011 was $14,988.75[13]. The simplest form of three dimensional display is the anaglyph. Anaglyph glasses have a red filter over one eye and a blue or cyan filter over the other eye. This type of display will allow the stereo data to be presented by sharing the three (red, green and blue) channels of a traditional display. The primary disadvantage to this system is that it is unable to display full color images in three dimensions.

Figure 3 Stacked Projectors

Stereo Photography

The human stereo visual system works because we have two eyes and there is about a 66 mm separation between the eyes [14]. In order to produce stereo photographs we need to produce a separate right and left eye image. This can be done by using a single camera with a lens for the right and left eyes as seen in Figure 4. This camera was called the stereo realist and was produced by the David White Company from 1947 to 1971[15]. The camera used slide film and the transparencies were mounted in special cardboard mounts shown in Figure 5. The mounted slides could then be viewed with a stereo viewer. The stereo viewer shown in Figure 6 operated much like a ViewMaster viewer. Stereo photographs can also be produced using a single camera if that camera is shifted to the right or left. This procedure works best if the shift is horizontal with no vertical component. While the camera can just be moved this method will not produce consistent results. In order to produce consistent results the camera can be attached to a slide bar for macro photography. One such device is made by Manfrotto (model number 3419) and is show in Figure 7. The camera and slide bar assembly can then be placed onto a tripod and a digital camera can be used to capture stereo images. The slide bar has a locking screw in the front which is used to lock the slide control in place. If this screw is not tightened down the camera may exhibit vertical movement between successive shots. Figure 8 shows a digital camera set up for stereo photography. Figure 9 shows a pair of photographs taken with a Canon T2i and an EF-S 18-55 lens set at a zoom of 36mm and there is 35mm of separation between the two images. Figure 10 shows the two color images merged together. The merged image was then converted to black and white in Photoshop and levels was used to set the output of the green and red channels on one photograph to zero and the output of the green and blue channels to zero on the other photograph as shown in Figure 11. The camera was initially focused on the tree in the foreground, the lens was turned to manual focus and image stabilization was turned off.

The exposure was manually set at f/20 for 1/50 second which was the camera's metering of the first exposure.

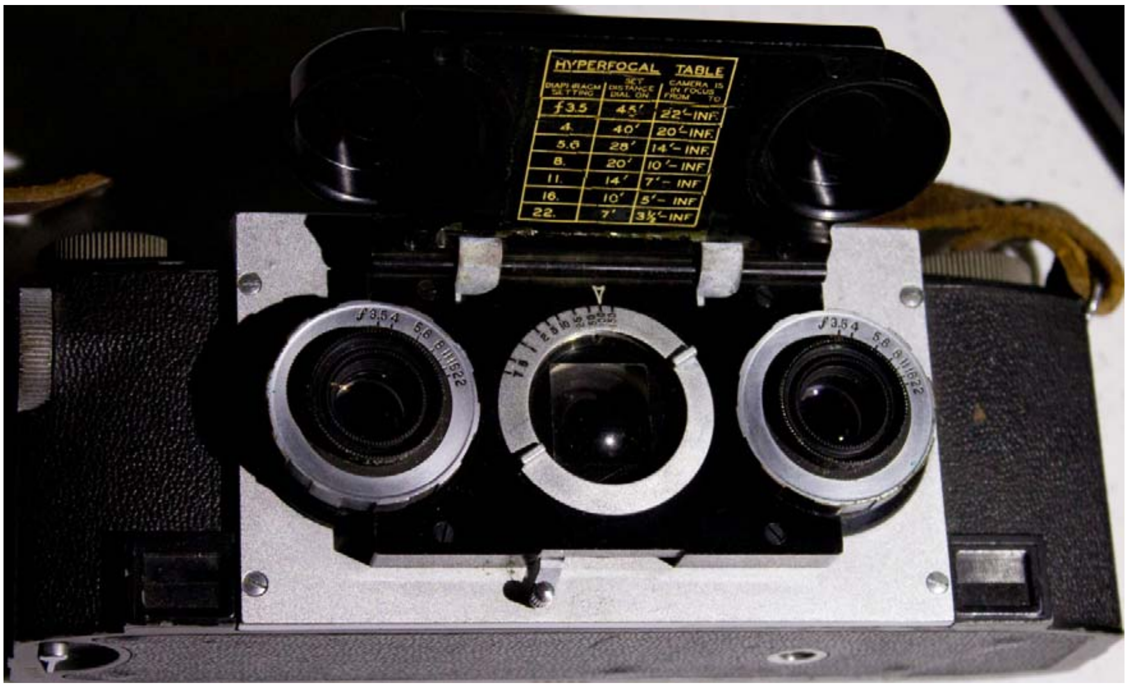

Figure 4 Vintage 3D Film Camera

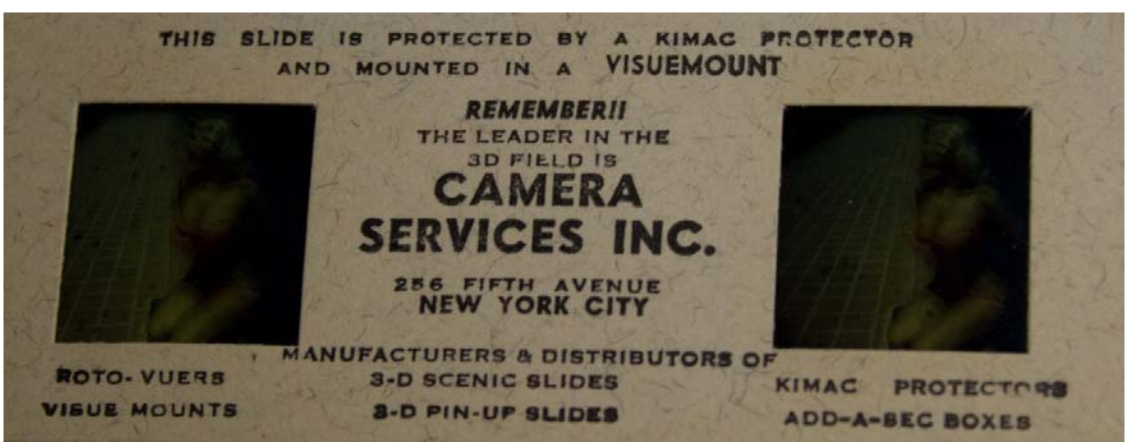

Figure 5 Stereo Realist Mounted Slide

Figure 6 Stereo Viewer

Figure 7 Stereo Slide Bar

Figure 8 DSLR Stereo Setup


Figure 9 Left and Right Image from DSLR

Figure 10 Merged Left and Right Images

Figure 11 DSLR Anaglyph

GeoWall [16]

The GeoWall consortium is a group of academic institutions that have shared software and hardware expertise to build a display wall capable of showing three dimensional imagery. The basic display properties of my visualization system and the GeoWall are the same. Both systems use passive stereo display through two projectors with polarizing filters. The GeoWall uses a single computer with a two video outputs to drive the projectors. The software in the GeoWall is a collection of stand-alone programs designed to perform a single task. This means that adding to the Geowall software involves writing a piece of software which does not have to be integrated into an existing system. The Geowall system has been tested with a number of projectors and has a metal stacking cabinet which can be used to house the projectors. Since the GeoWall project is a collaboration of academic institutions the available software will run on a variety of hardware platforms. Some of the software is Windows specific and other packages will run on either or both of Apple platforms or Linux platforms. Geowall has a number of image viewers available. One of the viewers available is called Pokescope[17]. Pokescope comes in two different versions, standard and professional with current pricing at $29.95 and $79.95. Both versions have implemented a slide show capability, conversion of anaglyph images to other formats, and swapping left and right images. The programs allow loading stereo images from one or two files (it will not allow the loading of separate files for each of the red, green and blue channels) nor does it perform and image processing functions on the images. Another software program for viewing stereo images is called Viewer[18]. This software is a viewing program for stereo pairs. This package also allows for a multi-file slideshow capability. The GeoWall project can also use commercial software that supports side by side stereo. One such package is ArcScene[19] from ESRI. ArcScene will allow three dimensional viewing from within ArcGis. GeoWall will also support the stereoview plugin to Matlab[16]. A software package called StereoGIS[20] can also be used with the GeoWall. This package allows the user to display and interact with DEM files as well as shape files. This package, like ArcScene, would allow the user to view, extract and manipulate terrain data. Z-Flux[21] is a package which allows the user to display three dimensional stereo animations. The software supports stereo viewing with side by side stereo. Z-Flux is available free of charge. The Geowall project can display three dimensional imagery with any package capable of side by side stereo display.

GGobi[22]

GGobi is the descendant of a package called XGobi. This package is a data visualization system. GGobi uses data from a single file in XML format and has a limited capability to extract data from files in XGobi format[23]. XGobi used a number of files to create a dataset. The files had a common base name and stored data in one file and labels were stored in other files. GGobi presents the user with data in both Cartesian and parallel coordinates. GGobi uses any number of separate windows to display the data and a window may contain multiple views of the data. The separate display windows can be linked so that a change in the representation of the data in one window can be observed in the other windows. GGobi presents the user with a number of ways to view data including: scatterplots, scatterplot matrices, parallel coordinates, time series plots and barcharts. GGobi allows the user to conduct tours of the data, as well as brushing data. GGobi can also directly manipulate the data using its display windows. GGobi can be used with R. R can be used to manipulate the variables within GGobi.

Wegman Patent

In 2002 Dr. Edward J. Wegman was awarded patent number 6448965 [24] for developing the idea of a voice controlled immersive virtual reality system. This dissertation develops technical details for the implementation of several aspects of that patent. The patent relies on voice input for user control of the virtual reality system. The visualization system of this dissertation could use voice as a supplemental input control without a great deal of effort.

GMU Software

The Center for Computational Statistics has developed a number of software packages for data visualization. These packages were all designed to handle extremely large data sets and were developed during a time when memory was an expensive commodity. Following is a description of the packages.

Mason Hypergraphics, © 1988, 1989 by Edward J. Wegman and Masood Bolorforoush, a MS-DOS package for high-dimensional data analysis.

Mason Ridge, (c) 1990, 1991 by Qiang Luo and Edward J. Wegman, a UNIX package for Silicon Graphics workstations for two- and three-dimensional density rendering using stereoscopic displays, transparency and lighting models and for ridge estimation.

Mason Hypergraphics, © 1991, 1992 by Ji Shen and Edward J. Wegman, a UNIX package for Silicon Graphics workstations for high dimensional data analysis using parallel coordinates, parallel coordinates density, and the density grand tour.

ExplorN, © 1992, by Daniel B. Carr, Qiang Luo, Edward J. Wegman, and Ji Shen, a UNIX package for Silicon Graphics workstations incorporating scatterplot matrices, stereo ray glyph plots, parallel coordinates, and the *d*-dimensional grand tour.

3-D MRI, © 1993, by Qiang Luo and Edward J. Wegman, a UNIX package for Silicon Graphics workstations for reconstructing 2-dimensional MRI slices into a 3-

dimensional voxel image with the capability of rendering iso-density contours in transparent stereoscopic display.

CrystalVision, a package written for Windows NT which can display data sets in parallel coordinate or 3D scatterplots. The package can do generalized grand tours as well as presenting the user with an array of 2D scatterplots from the different axes.

CHAPTER THREE

System Design

There will be three distinct classes of computers which comprise this visualization system. The first class consists of a single computer which is referred to as the "controller". This computer provides the primary user interaction and orders the other computers to produce output on their displays. The display on the controller is assumed to be minimalistic. This would allow the controller to be a generic computer with either no display or a very small size or resolution as compared to the system's overall display capability. A design constraint of the system is to allow the single controller computer to run the both the control and display subsystems thereby allowing the system to be debugged on a single computer which would make development on a laptop possible. This debugging capability is true concerning the major software components, but it may not be possible to debug every aspect of the software using only a single computer. The other class of computer is responsible for displaying imagery or data. The display computers do not have a specific hardware requirement. However, in order to display stereo three dimensional data the display computer will need to be using a projector with a polarizing filter. It is possible to create a display subsystem which has both stereo capable display and more traditional (non-stereo) display hardware. The stereo display computers will always occur in pairs, one computer each for the right eye image and another for the left eye image. The system currently utilizes only two computers for the display subsystem, but this could be expanded to include multiple projectors as well as any number of LCD panels. During system initialization, each computer in the display subsystem will connect to the controller computer. The controller computer will remain in initialization mode until all of the computers in the display subsystems have checked in. When each of the display subsystem computers check in, the controller passes the details of the display configuration back to each display computer. The display node computers never communicate amongst themselves, communications only occur between the controller computer and display computers. The details of the display configuration include the size of the virtual display that each physical display is a part of, which pixels of the virtual display the physical display is responsible for showing and the row and column location of the physical display. The IP address of each display subsystem computer is used to determine the location of that computer's display within the virtual display. There is no requirement that the displays are physically contiguous. Since LCD panels have bezels, a number of pixels may separate the end of one display and the beginning of the next display. The display configuration is set statically in the controller computer and is done with a table of values. Communication between the computers may occur in both directions but primarily the communications consist of orders from the controller computer to the display computers and short responses from the display computers back to the controller computers. The system assumes that each computer has access to the same file system. While this could occur by each computer sharing a single file networked disk volume, currently all the files are duplicated on each computer within the system. This system is designed to be able to accommodate both a stereographic display system and a large non-stereo display subsystem. While parts of the code base have both the display subsystems integrated, other parts have yet to have both subsystems implemented. The visualization system is best understood as a stereo display system with one projector for each of the left and right eye images which can be easily expanded to a stereo display system with multiple projectors for each eye and with a large multi-monitor non-stereo display.

Communications system

Communications occur between the systems over an IPv4 network. Each computer uses a socket to send and receive TCP packets between itself and the controller computer. The controller sends its instructions to the display computers via a set of predefined command numbers. Each packet sent consists of the command, a fixed block of flags, the length, in bytes, of the payload and the actual payload as seen in Figure 12. The communications system supports the commands summarized in Table 1. The communications system drives both the controller computer and its display clients. The controller and display processes first perform initialization and then enter a loop where visualization messages received through the socket are processed and then Windows messages are handled. This loop terminates only when communications are broken between the controller and its clients during a system takedown. Communications are constantly polled during the execution of controller and client processes to see if messages exist which need to be handled. The controller computer can include flags which may tell the client computers to send an acknowledgment packet back. The controller computer will change its cursor to the Windows “spinning hourglass” as long as there are outstanding acknowledgments to be received. This acknowledgment packet is sent back only at the completion of the requesting command. For example, if the client receives a request to display an image the acknowledgment is sent only after the image is actually displayed, not when the request is received. The acknowledgment is not used to indicate the success of an operation. If the client were unable to locate the image file to be displayed the acknowledgment would still be sent since the command was successfully received.

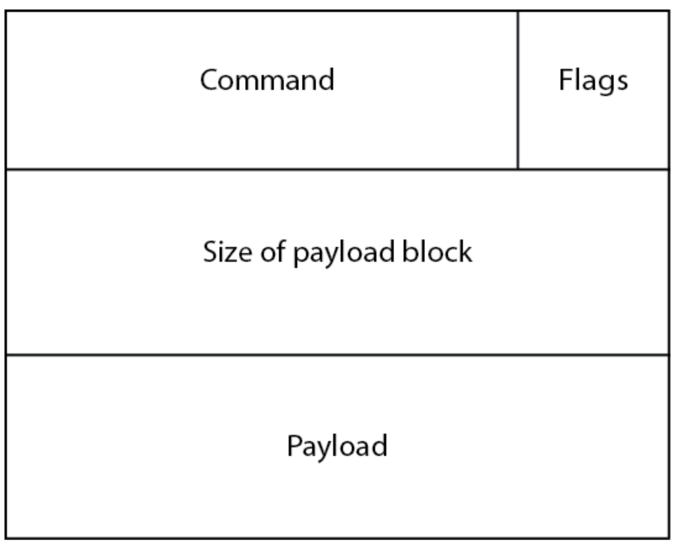

Figure 12 Communications Data Structure

FNAME	SETPOLYMODE	POINTSIZE	SPIN3D
PIXZOOM	RGBFNAMES	MOUSETRKON	NEXTSPIN
QUIT	RGBEND	MOUSETRKOFF	PARAXES
EDGE_DCT	IMGSEQ	AGIF_FREERUNLOOP	SCATTER3D
MINIMIZE	CENTER_IMAGE	AGIF_FREERUNNOLOOP	SCATTER2D
RESTRE	PAN_PIXELS	AGIF_STOP	LOCKAXESSCALE

REDRAW	IMGSEQ_START	AGIF_DISPLAYFRAME	UNLOCKAXESSCALE
MMARK	IMGSEQ_END	USE_VERT_AXES	CARTXAXIS
ECHO	RECENTER	USE_HORIZ_AXES	CARTYAXIS
ECHORESP	DATASETSTART	RESETAXES	CARTZAXIS
BLOCK	AGIF_NEXT	CUBE	CARTAXREQ
MOUSESPOS	REQ_FRAMECNT	CAMREQ	ACKRSP
MOUSEPOS	HIDE_AXIS	CAMUPDATE	AXSTATRSP
MOUSEEPOS	SHOW_AXIS	LOTTERYPICK3	FRAMECNT
MOUSECLR	DATA_COLORBYAXIS	LOTTERYPICK4	AXISORDERRSP
HISTOGRAM	NEXTAXES	LOTTERYCASH5	WAIT_RSP
CROP	PREVAXES	LOTTERYWINLIFE	CAMRSP
PCAPFILE	AXSTATREQ	LOTTERYMEGAMIL	CARTAXRSP
EIGHTBIT	AXISLABEL	MDLVIEWIDENT	
HISTMODE	AXISORDERREQ	MDLVIEWUPDATE	

Table 1 Communications Command Set

Software Components

The software of the visualization system consists of several distinct subsystems. The subsystems can be grouped according to the type of data that is going to be displayed. The two major types of data being displayed are either images or data. The image display system will be referred to as the image viewer and will cover all types of image data display. The data display system is responsible for all types of display that are not image related. The distinction between these two types of data display is somewhat arbitrary. For example, video would be handled as a component of the image viewer, and a histogram of an image is also handled by the image viewer. Image display is done in a two dimensional mode, even for stereo images. The three dimensional effect is the result of each of two two-dimensional images being displayed at the same time to create the illusion of depth.

Image Viewer

The image viewer section is responsible for the display of image files on the monitor. These images may be either three dimensional images or two dimensional images. Each stereo display node is predefined to assume the role of either the left eye or the right eye. However, this designation is only transitory, since some stereo images don't always follow formatting conventions. The viewer works by ingesting an image, in whatever form it may currently reside, and converting it into the pixels composing the image. The bitmap of pixels can then be manipulated to display the desired result on the projectors. The images displayed can be in any of the popular formats such as jpeg or gif. The images are opened using the Microsoft CImage class. This class will open any image in jpg, gif, bmp or png format. Images in more exotic formats, such as files in NASA's Planetary Data System (pds) format can also be opened and displayed. Other image formats can easily be added and integrated into the image viewer. The display processes first convert from the image file into an internal bitmap form. This internal bitmap is then used to fill the frame buffer for the display hardware.

Separate Files

Images for the right and left eyes can be stored as separate files and these file names are then passed to the image viewer for display. There is no standard designation for right and left eye image files (other than putting "right" or "left" into the file name). Since a color image can be composed of its three color channels (red, green and blue) each of those channels can be placed into a separate file. Thus a single stereo image may be held in six separate files. The controller computer will pass along a special flag to the display computers that a series of file names will be coming, and that an image should not be presented until the entire set of file names have been processed. The display computers will zero out any channels that do not have data passed in for them. Since the display system can display a file for each of the right and left eyes, the user can use the same file both the right and left eyes and be able to view regular (non-stereo) images.

Anaglyph Images

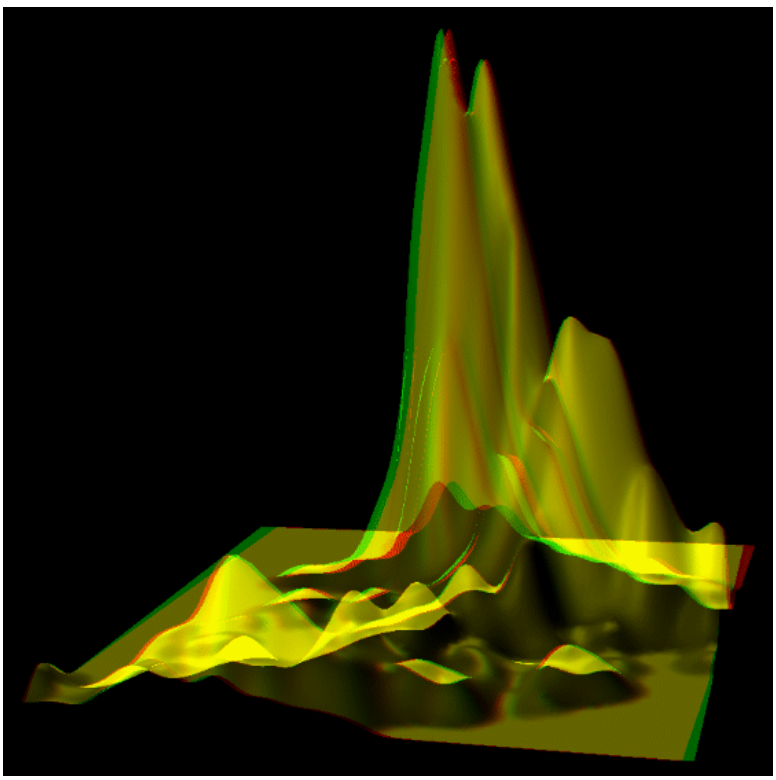

Figure 13 Anaglyph Example

An anaglyph image is a three dimensional image which separates the right and left eye images by means of color. Most images use red for the right eye, and green, blue or cyan for the left eye. Because we will have a single display device for each eye, we can use the right eye computer to display only the red channel and the left eye computer to display the other two colors. Further, since we have a dedicated display device for the right eye, we can display that image as a grayscale image by replicating the red value into the green and blue channels. The left eye image can also be manipulated in a similar fashion but not if both the green and blue channels have useful information. The image for the left eye is examined on a pixel by pixel basis and a grayscale value for each pixel is determined by looking at the green and blue channels. If the values in the green and blue channels are the same this value is replicated into the red channel thus establishing the grayscale value. If either the green or blue channel is zero, then the grayscale value is determined by the non-zero channel. If the values in the green and blue channels are different and both are non-zero then the grayscale value is determined by the average of the green and blue values. The example image was produced by combining two images taken with a digital camera. The two images were taken with a separation of 60mm at f/20 with a zoom lens set to 28mm. The two images were imported into Photoshop and converted to layers. Each of the layers was converted to black and white using the standard Black & White Conversion. The final image is very dependent on this conversion. There are presets for conversion using high contrast red and blue filters. The conversion using these high contrast filters produces a very saturated result. The levels command was used to set the green and blue channels' output to zero on the right eye image and to set the red and green channels' output to zero on the left eye image. Each of the resultant layers' blending mode was then set to screen.

JPS Images

Figure 14 JPS Example

JPS files (JPEG Stereo) have both the right and left eye images in one file, with the right and left images beside each other. The image viewer, by default, treats the image on the far left as the right eye image. The right and left eye images are assumed to be exactly the same size and that there is no border between the two images. If this assumption is correct, the number of horizontal pixels in the set of two images should always be a multiple of two. Should there be an odd number of horizontal pixels, the center column of pixel width one is discarded.

Animated GIF

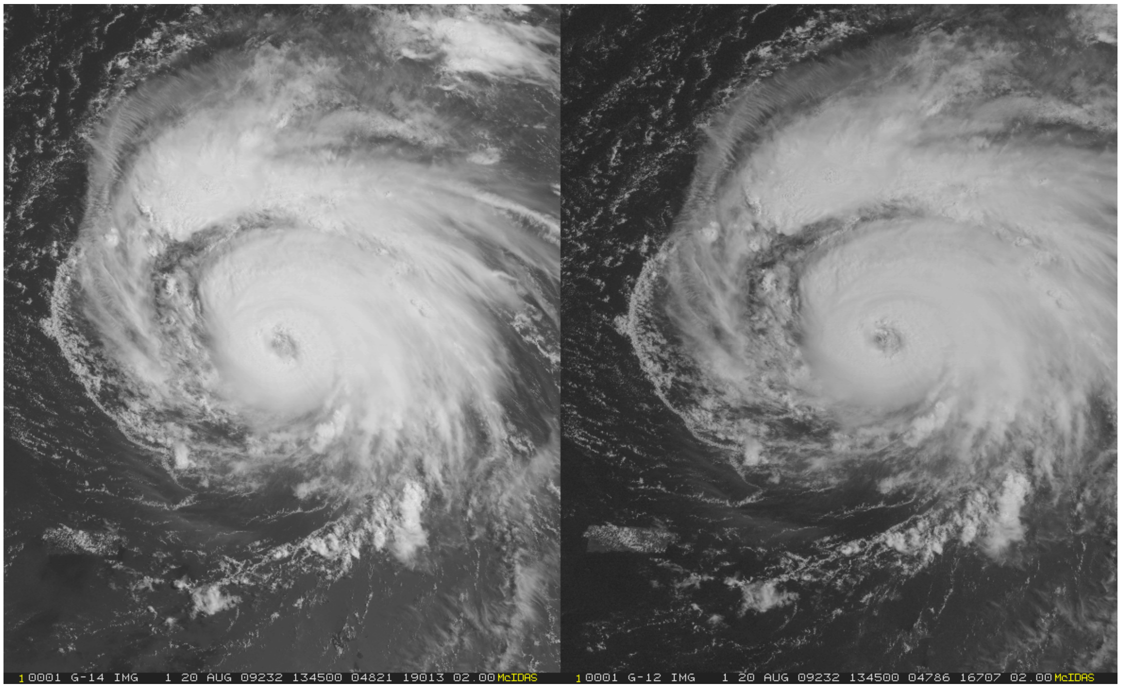

Figure 15 Animated GIF Image Courtesy **Cooperative Institute for Meteorological Satellite Studies (CIMSS), University of Wisconsin – Madison, USA[25]**.

An animated GIF file can be thought of as a small movie. The file contains a relatively small number of frames. The process of playing the frames in sequence produces the same effect as a flip book. The images contained in the animated gif may be in a stereo format which will produce a three dimensional animation. The three dimensional animation will require the individual frames to be presented in sequence and with each of the right and left frames presented simultaneously. The simultaneous presentation is accomplished by having the display computers inform the controller after an image has been displayed, and then the controller is able to then send the display computers an order to display the next frame in sequence. The primary difference between this type of file and a video file is that in a video file each successive frame is stored as a difference of pixels from the last frame. Each successive frame in an animated GIF file is a complete image.

Mars Rover

NASA landed two robots on the surface of Mars in January 2004[26] named Spirit and Opportunity. These two robots were equipped with digital cameras which have sent back a large number of images from the Martian surface. The images are in a standard jpeg format. Because the images are in a jpeg format the file name is used to provide catalog information on the circumstances of the image capture. The cameras on the rover produce gray scale images. Two of the cameras have a filter wheel that allows full color images to be produced by taking consecutive images through color filters. These two cameras are mounted to allow full color stereo images to be produced. Unfortunately, the filter wheels were designed to produce scientific images and the filter choices were not the best for producing visible images. The left eye camera has no green filter, thus all full color stereo images are missing this one channel. The rovers have a set of stereo cameras on the bottom of the rover forward and aft, two sets of stereo cameras mounted on a pole above the rover, a microscopic imaging camera and a camera to cover entry, descent and landing. This gives each rover a total of four sets of stereo cameras and two additional non-stereo cameras.

Mammography Images

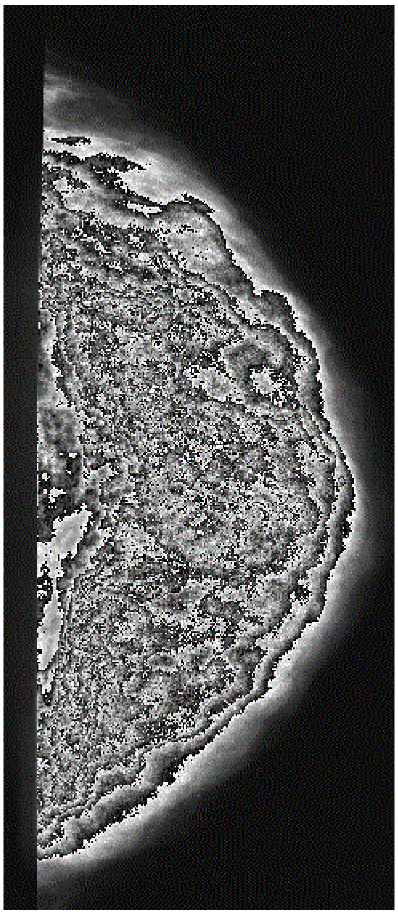

Figure 16 Mammogram

The visualization currently supports two collections of mammography databases. The two databases are the DDSM collection from USF and what appears to be the Nijmegen collection from the Netherlands. This second set is no longer easily found on the Internet. This second set consists of the mammography images as well as a second file which marks the location of anomalies in the image as identified by an expert. The collection of DDSM files are only mammograms, but have a number of different volumes. Each of the volumes share common characteristics, such as cancerous tumors, no tumors or benign tumors.

Visible Human

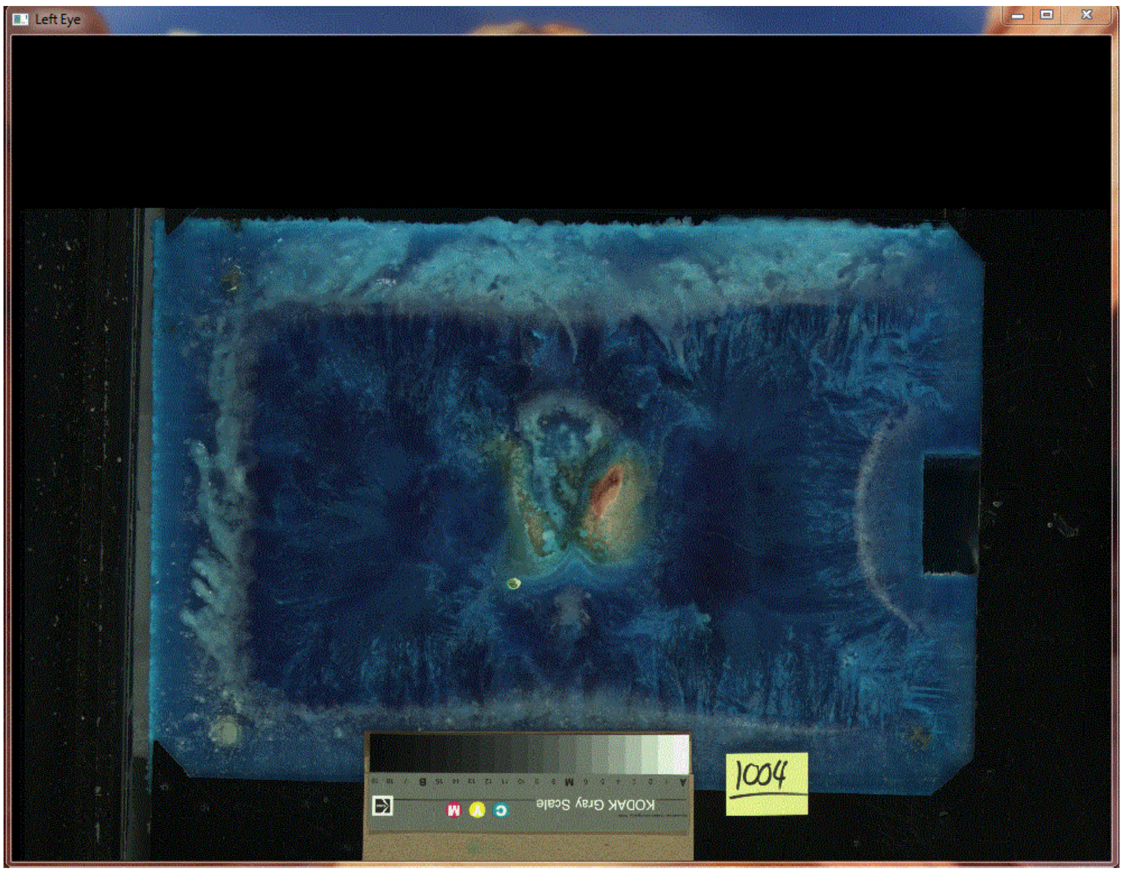

Figure 17 Visible Human Color Image

The Visible Human dataset is a collection of anatomical data from the National Library of Medicine. The data was gathered from a male and female cadaver and consists of full color images as well as MRI and CT data sets. The female data set is higher resolution than the male data set. The full color images are pictures taken as the frozen cadaver is removed horizontally one small slice at a time.

Data Files

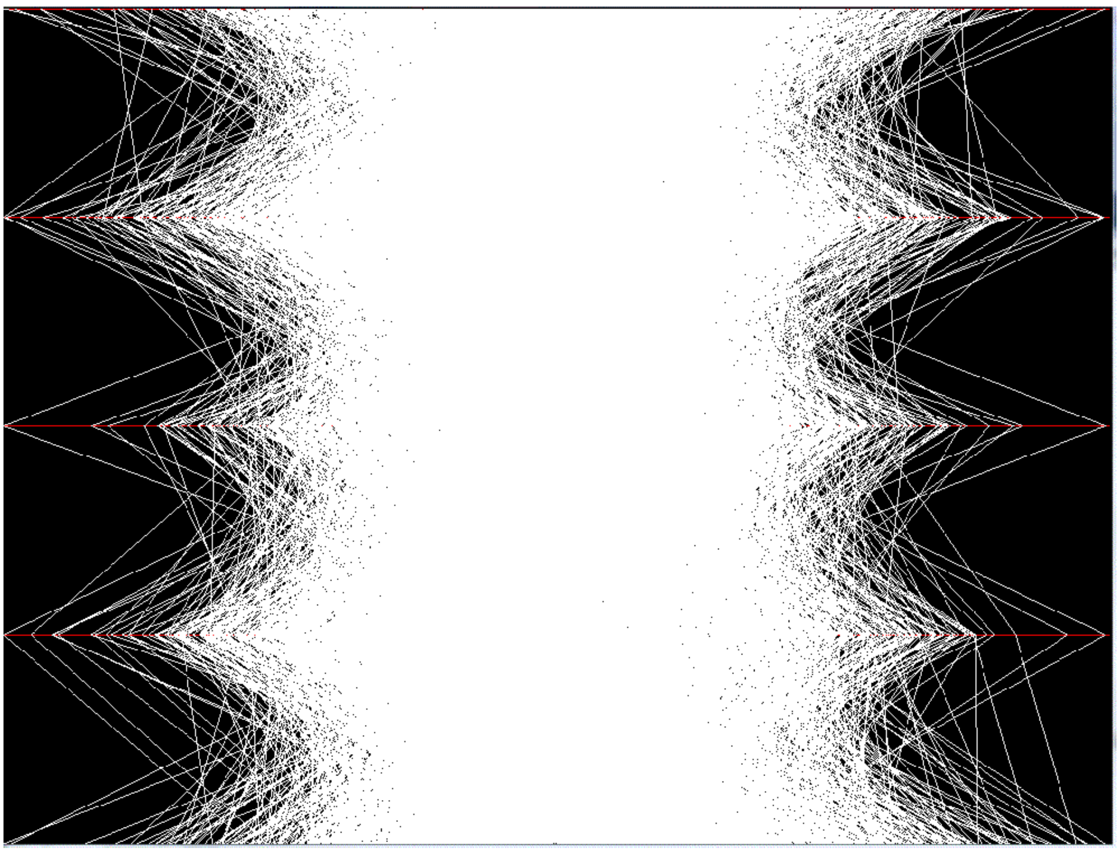

Figure 18 Parallel Coordinates

In addition to imagery data, the visualization system will display multi-dimensional data sets. The data is displayed in parallel coordinates or as a scatter plot in Cartesian coordinates. The parallel coordinates method of display will allow an arbitrary number of dimensions to be displayed simultaneously while the Cartesian coordinate system is limited to a maximum of three dimensions displayed simultaneously. The Cartesian plots are in either 2D or 3D. When a 3D scatterplot is requested, it is understood to be a stereoscopic 3D perspective view. There are a number of formats available to display the data sets. The primary format is a data file used by CrystalVision. An alternate form of data input is the comma separated value form. There is also an option to display data from the Virginia lottery.

Network Data

There is a large dataset of IP header traffic which can be displayed using the visualization system. This data was collected from the GMU campus and has only header information. This dataset is somewhat dated but is used to show the capability of displaying data with an order of magnitude in the terabyte range.

Tools

There are a number of tools available to enhance the visualization system. The available tools include a network packet timing system which allows an estimate of the time required to pass a packet from the controller computer to the display computers and back again.

CHAPTER FOUR

Implementation Details

The software has been developed in C++ and currently exists as three separate programs, one for the controller computer and one each for the right and left eyes. These three programs were initially one single source file with conditional compilation directives to handle the different aspects of the program specifics. This quickly became a very difficult workflow to maintain as the source base was really two different programs, one for the controller and one for the right and left eye. The code base was then split into three separate programs. Two of the three programs, the right and left eyes, differ only in the setting of a variable to indicate whether it is the right or left eye. These three programs contain two large C++ classes, one to handle communications and one to handle the display graphics. These two classes still use conditional compilation as these two classes share so much code between the three programs. There isn't a bright line between what is handled in each of the two classes. The software was created and tested with Microsoft Visual Studio under Windows XP and Windows 7 as well as being used on both 32 and 64 bit hardware.

Initialization

The control process begins by creating instances of the display and communications classes. This will create a graphics window, but it is currently unutilized. Visual Studio will create the menu bar and other interface resources used by the software. The control process is responsible for user interaction. The display processes are assumed to exist without the capability of user input. During the initialization of the communications class, a data structure is built to maintain control of the communication sockets. This data structure keeps track of the sockets as well as details such as if it’s a right or left eye and screen resolution. The data structure format is shown in Figure 19. The client process begins with the initialization of the communications and display classes. The display initialization will create an OpenGL window of predetermined dimensions. The client will save only the socket used to communicate with the control process.

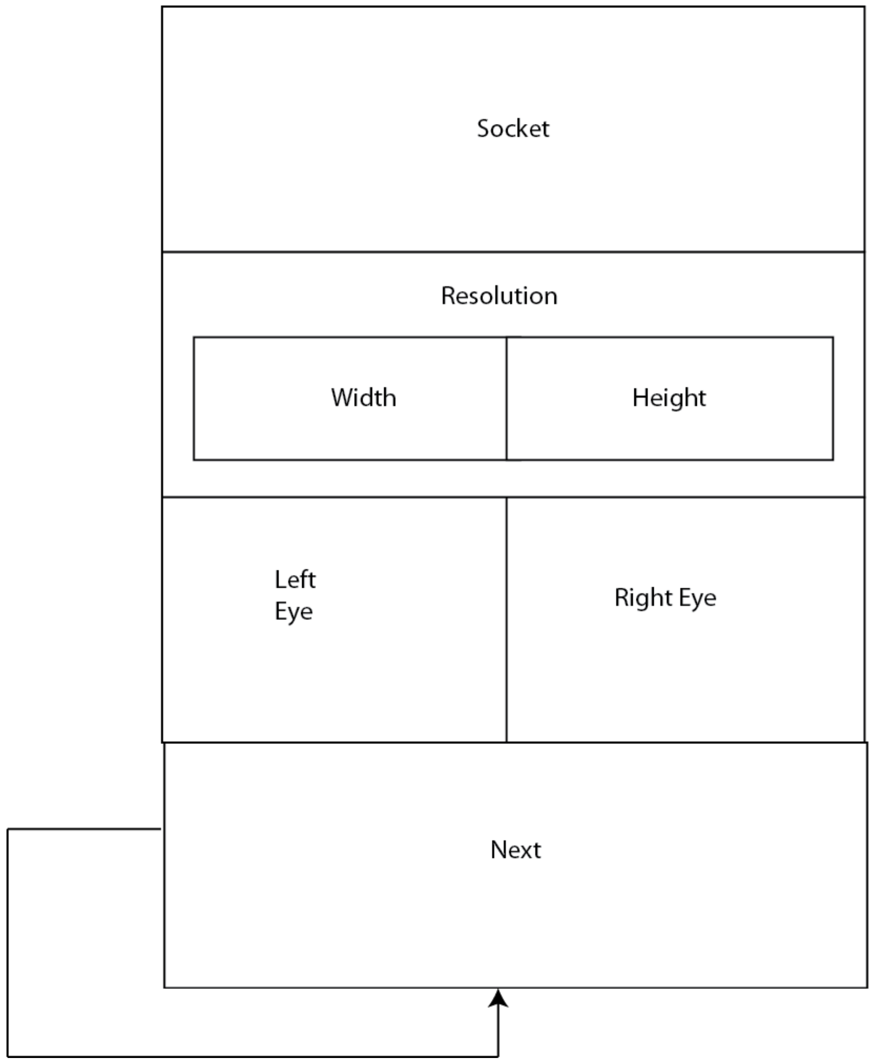

Figure 19 Display Node Data Structure

Display Class

The display class is responsible for all graphics activity. The graphics are all done in OpenGL. The class instantiation procedure sets up any variables which need default values later in the program. This class will also open the graphics window. The size of the screen being opened is set at compile time. The graphics window can appear in a windowed environment or can be run full screen. The full screen determination is done at compile time and cannot be changed during the execution of the program. The graphics window is opened in all three programs, but is never written to on the control program. Gamedev.net[27] provided some of the underlying code for opening the display window and setting screen characteristics. The display class uses OpenGL as its graphics engine and operates in either a two dimensional display mode or a three dimensional display mode. When images are being presented on the screen they are drawn using pixel drawing routines. The drawing engine is also operating in two dimensional mode and is using an orthographic projection to display data. The graph drawing routines also use the orthographic projection as seen in Figure 20.

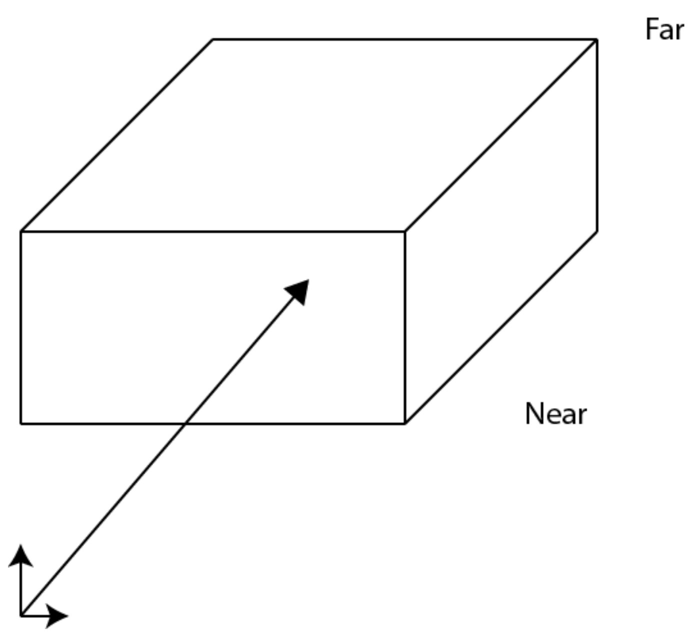

Figure 20 Orthographic Projection

Any object that are between the near and far clipping planes are projected onto the near clipping plane. The projection is done without a perspective transformation so sizes and angles are not distorted. The standard OpenGL perspective transformation is shown in Figure 21. This is a truncated pyramid and like the orthographic projection all geometry between the near and far clipping planes which is also within the interior of the truncated pyramid is shown on the near clipping plane.

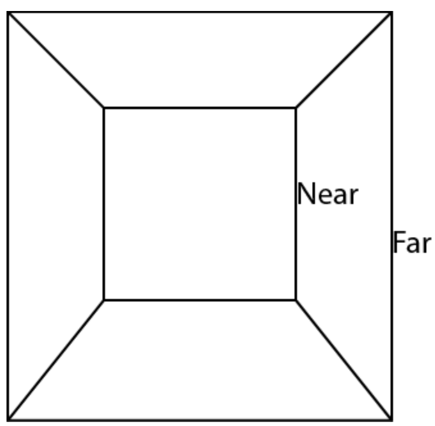

Figure 21 Perspective Transformation

This perspective transformation works correctly when there is a single monitor to display a three dimensional image. Because we have two projectors to display a three dimensional scene we need to use a transformation that is aware of the two viewpoints. Figure 22 shows the top view of the perspective transformation that is used[28].The center projection in this diagram is the standard perspective transformation. The right eye projection is the regular perspective transformation with shifted to the right. The left eye frustum is shifted in a similar fashion. Figure 23 is the same figure as the prior one except the left eye view has been removed for readability. In order to construct the frustum, we will need to know where the line segments emanating from the cameras intersect the "near" plane. The angle Q is one half the aperture angle and we will represent the corresponding angle in the right eye camera as R. That angle is not labeled in the figure for readability. We can use the following to derive the length of the frustum for the right side of the right eye by the following, assuming that the center camera is located at the origin:

$$N = distance\ to\ the\ Near\ plane$$

$$F = distance\ to\ the\ Focal\ plane$$

$$Tan(Q) = \frac{sep + X2}{N}$$

$$X2 = \big(N * Tan(Q)\big) - sep$$

This can also be seen by observing that in the triangle containing A1 and X2,X3,X4:

$$X2 + X3 + X4 = N * Tan(R)$$

$$X3 + X4 = sep$$

$$Q = R$$

$$X2 + sep = N * Tan(Q)$$

$$X2 = \big(N * Tan(Q)\big) - sep$$

$$Tan(R) = \frac{X2 + X3 + X4}{N}$$

$$X3 = sep - X4$$

$$Cos(R) = \frac{F}{A2}$$

$$A2 = \frac{F}{Cos(R)}$$

$$Cos(R) = \frac{N}{A1}$$

$$A1 = \frac{N}{Cos(R)}$$

By similar triangles using the triangle formed by A1 and X4 and the triangle formed by A2 and the line segment between the two parallel lined directly behind X4:

$$\frac{X4}{sep} = \frac{A1}{A2}$$

$$\frac{X4}{sep} = \frac{\frac{N}{Cos(R)}}{\frac{F}{Cos(R)}}$$

$$\frac{X4}{sep} = \frac{N}{Cos(R)} * \frac{Cos(R)}{F} = \frac{N}{F}$$

$$X4 = \frac{N * sep}{F}$$

We know that

$$X2 = \left(N * Tan(Q)\right) - sep, \qquad X3 = sep - X4$$

We can derive X2+X3, which is needed for the length of the frustum on the right side

$$X2 + X3 = \left(\left(N * Tan(Q)\right) - sep\right) + (sep - X4)$$

$$X2 + X3 = \left(N * Tan(Q)\right) - X4$$

$$X2 + X3 = \left(N * Tan(Q)\right) - \left(\frac{N * sep}{F}\right)$$

The right side of the right eye frustum is given by the sum of sep, X2 and X3:

$$sep + X2 + X3 = \left(N * Tan(Q)\right) - \left(\frac{N * sep}{F}\right) + sep$$

We further observe that the last equation drops the final two terms when sep is zero, that is, the final two terms are due solely to the stereoscopic projection. The left side is derived in a similar fashion:

$$X2 + sep = N * Tan(Q)$$

$$X5 + sep = N * Tan(Q)$$

$$X5 = X2$$

$$\frac{X6}{sep} = \frac{A3}{A4}$$

$$\frac{X6}{sep} = \frac{\frac{N}{Cos(R)}}{\frac{F}{Cos(R)}}$$

$$\frac{X6}{sep} = \frac{N}{Cos(R)} * \frac{Cos(R)}{F} = \frac{N}{F}$$

$$X6 = \frac{N * sep}{F}$$

For the right eye:

$$right\ side = \left(N * Tan(Q)\right) - \left(\frac{N * sep}{F}\right) + sep$$

$$left\ side = -\left((N * Tan(Q) - sep) + \left(\frac{N * sep}{F}\right)\right)$$

$$= -\left(N * Tan(Q)\right) - \left(\frac{N * sep}{F}\right) + \ sep$$

The left eye can be determined by using the equations derived above and observing that the left side of the left eye is the negation of the right side of the right eye and the right side of the left eye is the negation of the left side of the right eye:

$$left\ side = \ -\left(\left(N * Tan(Q)\right) - \left(\frac{N * sep}{F}\right) + sep\right)$$

$$= -\left(N * Tan(Q)\right) + \left(\frac{N * sep}{F}\right) - sep$$

$$right\ side = \ -\left(-\left((N * Tan(Q) - sep) + \left(\frac{N * sep}{F}\right)\right)\right)$$

$$= \left(N * Tan(Q)\right) + \left(\frac{N * sep}{F}\right) - sep$$

. We now need to map the frustum back to the shape of the physical display because we want the same aspect ratio for the frustum and the physical display. We can accomplish this by either changing either the vertical or horizontal dimension of the frustum. The most straight forward way is to change the vertical aspect of the frustum by multiplying it by the display height divided by the display width. If we wish to change the frustum’s aspect ratio by manipulating the horizontal dimension we can’t use the terms which involve the stereo terms as multiplying these terms would change the location of the focal plane, the plane of convergence.

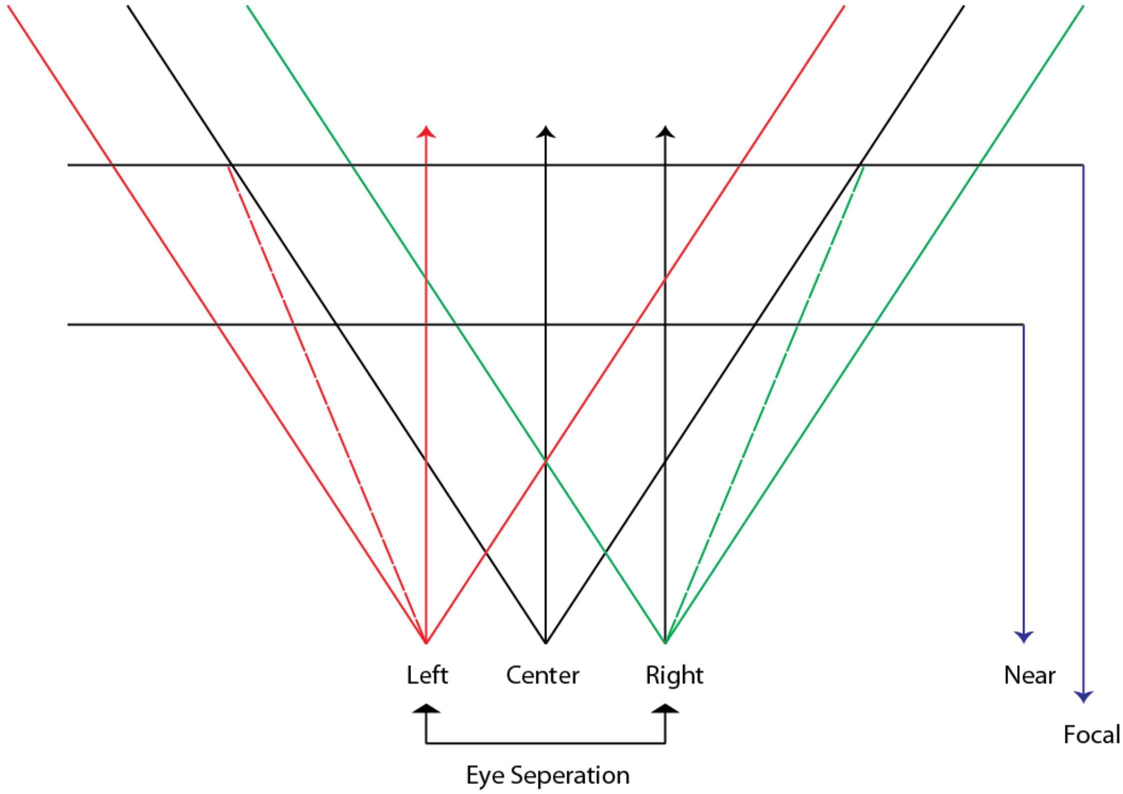

Figure 22 Stereo Perspective Transformation

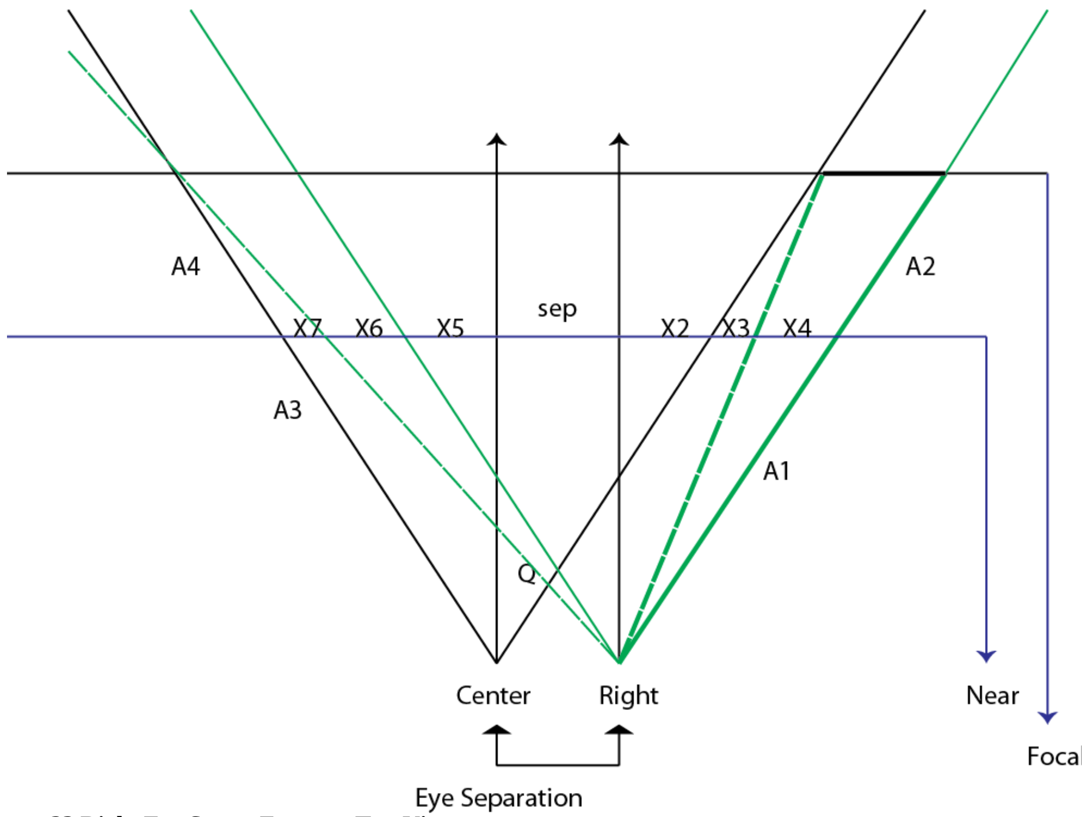

Figure 23 Right Eye Stereo Frustum Top View

These calculations are used to form a projection matrix. OpenGL will create a projection matrix of the following form[29]:

$$\begin{bmatrix} \frac{2N}{right-left} & 0 & \frac{right+left}{right-left} & 0 \\ 0 & \frac{2N}{top-bottom} & \frac{top+bottom}{top-bottom} & 0 \\ 0 & 0 & -(\frac{far+N}{far-N}) & -(\frac{2*far*N}{far-N}) \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

For example, if we wish to create a frustum with a 90 degree aperture, eye separation of 5, near plane at 50, focal plane at 100, far plane at 150 and screen size of 1024 x 768 pixels, using the prior equations we would calculate for the right eye the top is 50.0 and the bottom is -50.0. The right side is 69.17 and the left side is -64.17. Similarly, for the left eye, the top is 50.0, the bottom is -50.0, the right side is 64.17 and the left side is -69.17. The right eye projection matrix is:

$$\begin{bmatrix} 0.75 & 0 & 0.0375 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -2 & -150 \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

The left eye is:

$$\begin{bmatrix} 0.75 & 0 & -0.0375 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -2 & -150 \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

If we take three points in space, (10,10,-50), (10,10,-100) and (10,10,-150) in order to project them in a stereo plot we also need to move the left and right cameras from the origin to their correct locations which is given by translating along the x axis. Because OpenGL doesn't actually move the camera, we will move the points in space instead. This will result in evaluating the points (5,10,-50), (5,10,-100)and (5,10,-150) for the left eye. The right eye points are (15,10,-50), (15,10,-100) and (15,10,-150). Multiplying each of these points by the projection matrix for the left eye yields (5.625,10,-50,50), (7.5,10,50,100) and (9.375,10,150,150), with the right eye points being (9.375,10,-50,50), (7.5,10,50,100) and (5.625,10,150,150). We will now divide each 4-tuple by the fourth number yielding for the left eye: (.1125, .2, -1, 1), (.075, .1, .5, 1), (.0625, .067, 1,1) and for the right eye: (.1875, .2, -1, 1), (.075, .1, .5, 100), (.0375, .067, 1, 1). Because we are interested in the x and y coordinates we will drop the other two for now and notice that the difference in the x coordinates for the points on the near plane is

.075, there is no difference on the focal plane, and there is a -0.025 difference on the far plane. This indicates a positive parallax for the point in front of the focal plane, zero parallax on the focal plane and negative parallax on the point behind the focal plane.

Communications Class

The communications class is responsible for communications between the three programs. When this class is instantiated, a number of variables are set to default values and the communications sockets are set up. The control process will create a socket to listen on a port in user space. The software currently uses port 4242. The control socket will accept a predefined number of connections and must accept exactly that number of connections. The left and right eye processes will create a socket and attempt to open a connection to the control process. The control process's IP address is currently predefined in the software. The socket created is a TCP socket thus ensuring a reliable communications protocol. Once a communications channel is established between the control process and one of its display clients, the display process will send a packet identifying itself. The first packet sent is an integer which will be interpreted as meaning either right or left eye. This information is saved in the control program so it can direct messages either to the right or left eye. All sockets are created in blocking mode by default, but each one of the three programs will explicitly set this on their socket. The sockets can be switched to non-blocking mode programmatically. One of the more important flags that can be passed from the controller to the display process is the acknowledgement flag. When the control process sets this flag it also increments a master count of outstanding acknowledgement requests. The control process will indicate a positive value in this variable by changing its cursor. The display process will send an acknowledgement message once it has completed the action requested by the control process. The control process should not request an acknowledgement for actions which require an indeterminate amount of display frames. The control process does not stop the normal flow of processing while it is waiting for acknowledgements to arrive, however, the main processing loop is written to give preference to receiving and processing acknowledgement packets. The acknowledgement packets are a construct of the visualization system and are separate and distinct from the normal flow of TCP acknowledgements. Another flag which is processed by the controller's communication class is an instruction to wait for a response. This flag will force the controller process to not continue with the current flow of the program until the display computers respond.

Context

The control process does not need to send every message to each of the display nodes. The set of display nodes that messages are dispatched to is controlled by the context menu options. Most of the commands send their messages to the current context. The current context is by default set to the left and right display nodes. When the user requests to open a left and right file to display a stereo image, the current context is ignored and the file open requests are passed to the left and right display nodes. When a user requests that a stereo animated gif file be opened that requests is always passed to the right and left display nodes instead of the current context. If the user wanted to display a stereo animated gif in a single window then it would not be appropriate to use the menu option to open a stereo animated gif. The appropriate menu choice would be to open the animated gif as a flat file. If the requested file was an animated gif file in a jps format the file would display the two images side by side and show each frame of the animation. The socket blocking request is always passed to every socket within the visualization system. The quit command is always passed to every display node as well.

Main Processing Loop

Both the control and display processes share the same basic processing loop. These programs need to handle two distinct types of messages. Windows uses messages to inform the program of user interface events. The main processing loop of the programs must handle Windows events as well as communications between the display and control processes. The primary difficulty in the main processing loop pertains to needing to handle video issues. The primary mode of communication is to have the controller process send out a packet which requires the display process to show an image. There is a one to one correspondence between the packets sent and images displayed. However, in order to display video, or simply a multiple image gif, the client process must receive the order to start displaying images and not only display the images but possibly receive an order to stop the display of the image sequence. The processing loops of both the control and display processes work by constantly polling the status of inbound windows messages and inbound network traffic. The control process has a slightly simpler structure because it will never have to deal with unexpected messages from its display clients. The controller has a strong preference given to processing messages from the display clients over processing Windows messages. The control process performs two basic functions until the user requests termination of the program. First it will dispose of all messages waiting in the communications socket from the display clients and then if there are no outstanding acknowledgments waiting to come from the display clients it will process a Windows message if one exists. The two steps will repeat until the control process terminates. The processing loop in the display program behaves in the same basic function. The first step remains the same, all messages from the control program are processed. Then, a Windows message can be processed, if one exists. There are a number of Windows messages which the display program might need to respond to, such as a request to redraw the screen. Following this, the display program will display the next frame of an image sequence, if there is such an operation pending. This loop will continue until the display program requests termination, possibly as a result of processing a message from the control process.

Image Viewer

When the user selects a menu option in the control process, it will generate a Windows message indicating that the menu option has been selected. The control process will execute different code for all the different menu options, but as a basic rule, each menu option will result in a message being sent to the display units for action. When the control process receives a menu request to open an image file, it will open a Windows file explorer for the user to select an appropriate file. There is a menu option to select both a right and left image and if that option is selected the left image is chosen followed by the right image. Once an image is chosen the control process must set a number of flags to describe the content of the file. The flags are set based on the file name as well as actions reflected in the menu choice, but not the contents of the file. When the display process receives the request to display an image it will first examine the flags in the message and certain types of files will receive special processing. Files from the Visible Human project, mammography files and PDS files are all processed separately as are animated GIF files. If the file does not require special processing, it is passed to the Microsoft class for reading image files. There is no guarantee as to what format the image will be in after the file is parsed by the Microsoft libraries. The Microsoft libraries are able to parse most common image file formats. The image is then converted to an appropriate bitmap format. In the case of a JPS image, the original image is split in half and the correct half is displayed for each eye. If the original image was an anaglyph image, the right and left channels are separated and each of the two channels is then converted to a grayscale image. The software currently assumes that the anaglyph glasses are actually red/cyan glasses with red on the left eye. The bitmap format that is created is the size of the image, not of the display device. Currently, any scaling that is done is accomplished through the OpenGL scaling commands. This could easily be replaced with a custom written software solution. The software is written such that the original image is always stored as the parent bitmap and any changes that are written into the bitmap are written as a child bitmap. NASA uses a special format for some of their files called planetary Data System (PDS) files. These files are converted into a bitmap through the use of software supplied by NASA. Color image data from the Visible Human Project are stored as raw color images which have been compressed. These images are 2048 by 1216 pixels. The files are stored as sequential pixels in row format with each pixel being represented as three bytes. Radiographic data from the visible human project along with mammography data not from the DDSM project is processed as raw compressed grayscale files. The pixels are stored in row format and each pixel is two bytes long. The CT data files have a 3416 byte header and are 512 bytes by 512 bytes. This header is not currently used to extract data. The MRI data files are 256 bytes by 256 bytes and have a 7900 byte header. The mammography data files are 2048 by 2048 pixels and have no header information. The mammography files are further described below. The DDSM files are all stored in lossless jpeg format. The software for translating these files is from the Portable Video Research Group at Stanford[30]. Another type of file that can be displayed is a simple flat file. When the user selects that option, the file is opened and displayed by both the right and left eye display processes.

Mammography Files

The mammography files are grayscale images 2048 pixels wide and 2048 pixels high. These files do not have a header. The image files have a ".ima" file extension. Each of the image files has a companion file with a file extension of ".mrk". The companion file lists the anomalies in the mammogram. The format of the companion file is a series of integers listing the x and y coordinates of the anomaly as well as the radius of the affected area. The first number in the file is the total number of sets of the three descriptor numbers. Because the image is a grayscale image when the marker is shown, the green and blue channels of the image are set to zero in the affected area. Because a grayscale image has the exact same values in the red green and blue channels, the mark information can be removed by copying the red pixel values into the green and blue channels for each of the affected pixels.

Visible Human Files

The Visible Human files consist of CT, MRI and full color data sets. The MRI and CT files are the same basic format but with different header sizes. The radiological files are raw grayscale files preceded by a block of header data. The full color images are also in a raw format. All of these files are stored in a compressed form. The files are decompressed in memory and the uncompressed data stream is then converted into an image.

Animated GIF

An animated GIF file can be thought of as a small movie. This file format is based on the GIF format and supports multiple images within a single file. The animated GIF format is used to display images and the images may also have a format. The JPS format, while using JPEG in its name, doesn't have any requirement that the content be in JPEG format. Each frame of the animated GIF may actually contain a right and left image. The format is converted to a bitmap form with the use of Microsoft libraries. The control process will send a message to the display units that an animated GIF file has been requested. The control process itself will not actually open the animated gif file to determine its specifications. After the control process requests that an animated GIF be opened it will also send a request for the display processes to send information about the specifications of the animated GIF. The control process will then await the return of the animated GIF specifications. Once the specifications are returned, the control process will then present the user with a window to control the remainder of the process as seen in Figure 24.

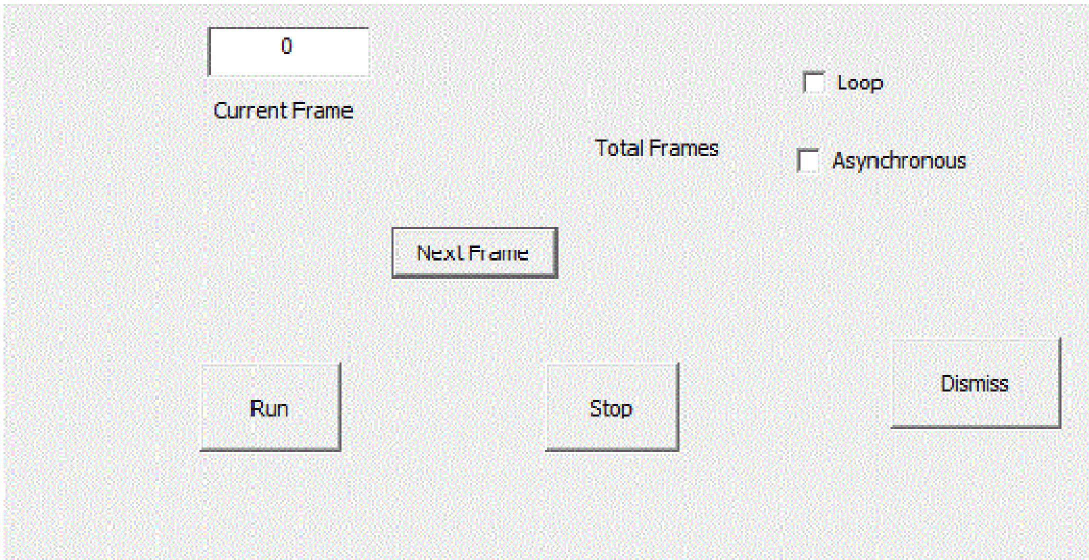

Figure 24 Animated GIF Control Dialog

The control process will then set the current frame to zero and report the total number of frames from the information sent back by the display process. The software will actually receive specifications from both the right and left eyes. Currently, the software will use the specification information that arrives last. The user can specify a frame to display in the single editable field in this window. If the user specifies a number greater than the total frame count the current frame will be set to zero. The effect of any change will be reported to the program only when the editable looses focus. The "next frame" button will advance to the next frame in the gif file. If the last frame was being displayed then the first frame will be displayed next. The "Run" button will cause the frames of the GIF image to advance without user interaction. If the "Asynchronous" box is not checked then the frames in the right and left eye display processes will display at (or near) the same time. If the “loop” button is not checked then the sequence of frames will stop displaying after the last frame has been displayed. If the user has requested the frames to display asynchronously then the current frame count will probably not be accurate but it will continue to increment to show that frames are being displayed. If the user elects to dismiss this dialog box then the display processes will terminate any sequences currently being displayed. Once the display process begins showing a sequence of images asynchronously the display is handled inside the main processing loop. Any communications from the control process will terminate an asynchronous display of image frames.

Mars Rover

The two Mars rovers, Spirit and Opportunity, landed on Mars in January 2004. These rovers have a number of cameras available to send Mars imagery back to earth. Each rover has right and left cameras on the front and back for hazard avoidance. These cameras are called the “hazcams”. Each of these cameras has a 120 degree field of view. There is a stereo camera for navigation. This camera has a 45 degree field of view. This camera is called the “navcam”. These cameras were all designed for engineering applications. The rover has a camera called the “pancam” for science applications. The pancam is a stereo camera and has a set of color filter wheels for each of the right and left cameras. Full color stereo images cannot be directly produced from the pancam because it doesn’t have a green filter on the right eye camera. Table 2 has the full specifications [31] for the rover’s filter wheels.

Left Filter	wavelength	bandpass	Right Filter	wavelength	bandpass
L1	739 nm	338nm	R1	436 nm	37 nm shortpass
L2	753 nm	20 nm	R2	754 nm	20 nm
L3	673 nm	20 nm	R3	803 nm	20 nm
L4	601 nm	20 nm	R4	864 nm	17 nm
L5	535 nm	20 nm	R5	904 nm	26 nm
L6	482 nm	25 nm	R6	934 nm	35 nm
L7	432 nm	32 nm shortpass	R7	1009 nm	38 nm longpass
L8	440	20 nm Solar ND	R8	880 nm	20 nm Solar ND

Table 2 PANCAM Filter Wheel

NASA provides the rover images in jpeg format after they have been radiometrically processed. Because these images are in jpeg format there is no header available to provide information about when the picture was taken. NASA uses the file name to provide the information about the imagery [31]. Table 3 summarizes the fields in the file name.

Character position	Description	comments
1	Spacecraft ID	1=Opportunity 2=Spirit
2	Camera	F=Forward HAZCAM R=REAR HAZCAM N=NAVCAM P=PANCAM

		M=Microscopic Imager E=EDLcam
3-11	Spacecraft clock	seconds since 1-1-2000 11:58:55.816 UTC
12-14	Product type	
15-16	Site number	
17-18	Drive number	
19-23	Command sequence number	
24	Camera eye	R=Right L=Left
25	Camera filter	
26	Product producer	
27	Version number	

Table 3 Mars Rover File Names

When the user selects the rover option in the menu instead of selecting a file to open the user must select a directory of rover files. The control process will then recursively descend through that directory and catalog all the image files it finds. The control program will maintain a sorted linked list of all the files as well as their locations. The control process will maintain a list of the directories where the files can be found rather than having the same text of the directory name appear in each file descriptor. The control program's data structure to maintain the file information is detailed in Figure 25.

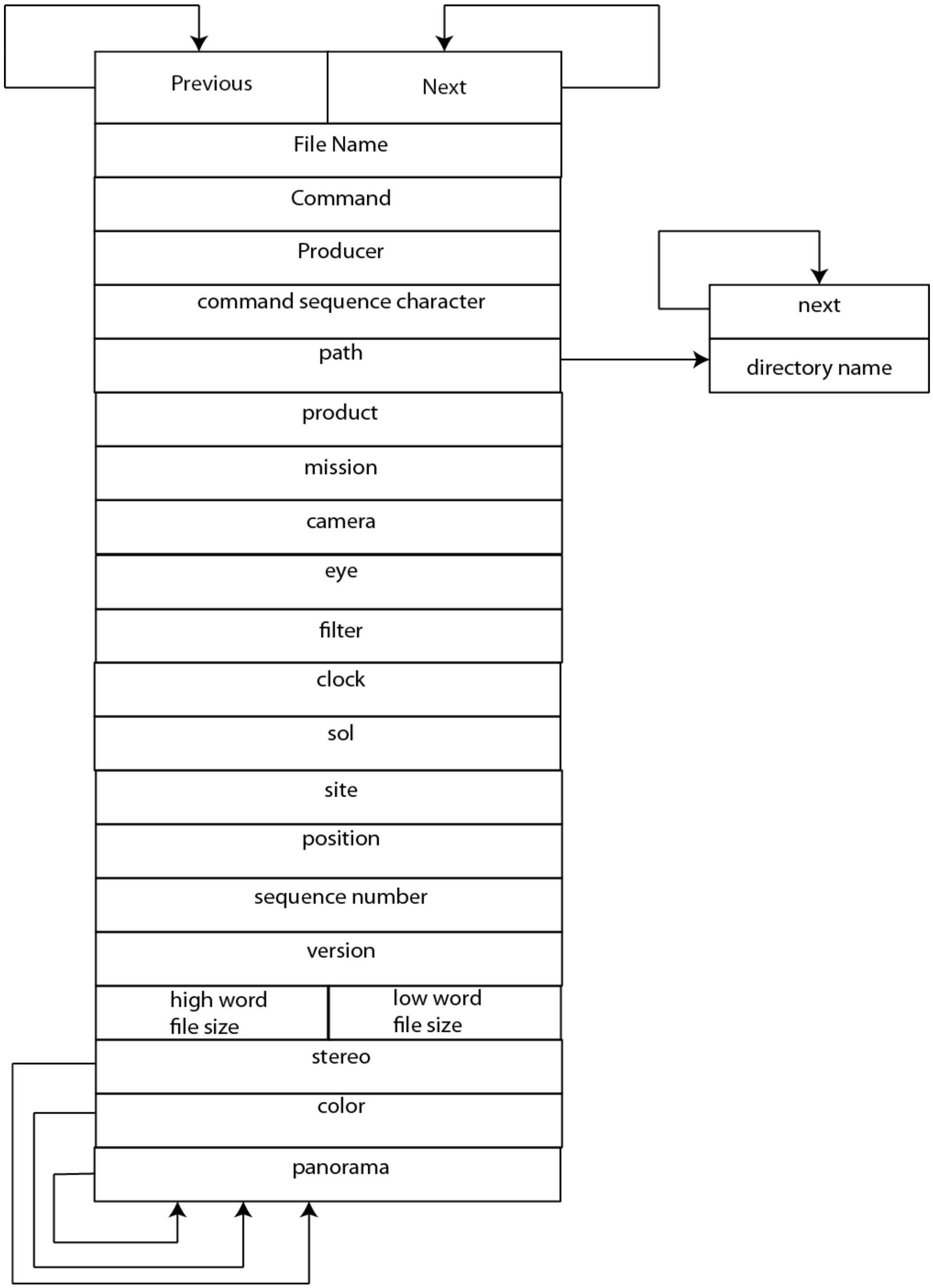

Figure 25 Rover Image File Data Structure

The control process only uses the file names to catalog the image files; it doesn't open the files to verify their content. Once the catalog is built the control process will display Figure 26.

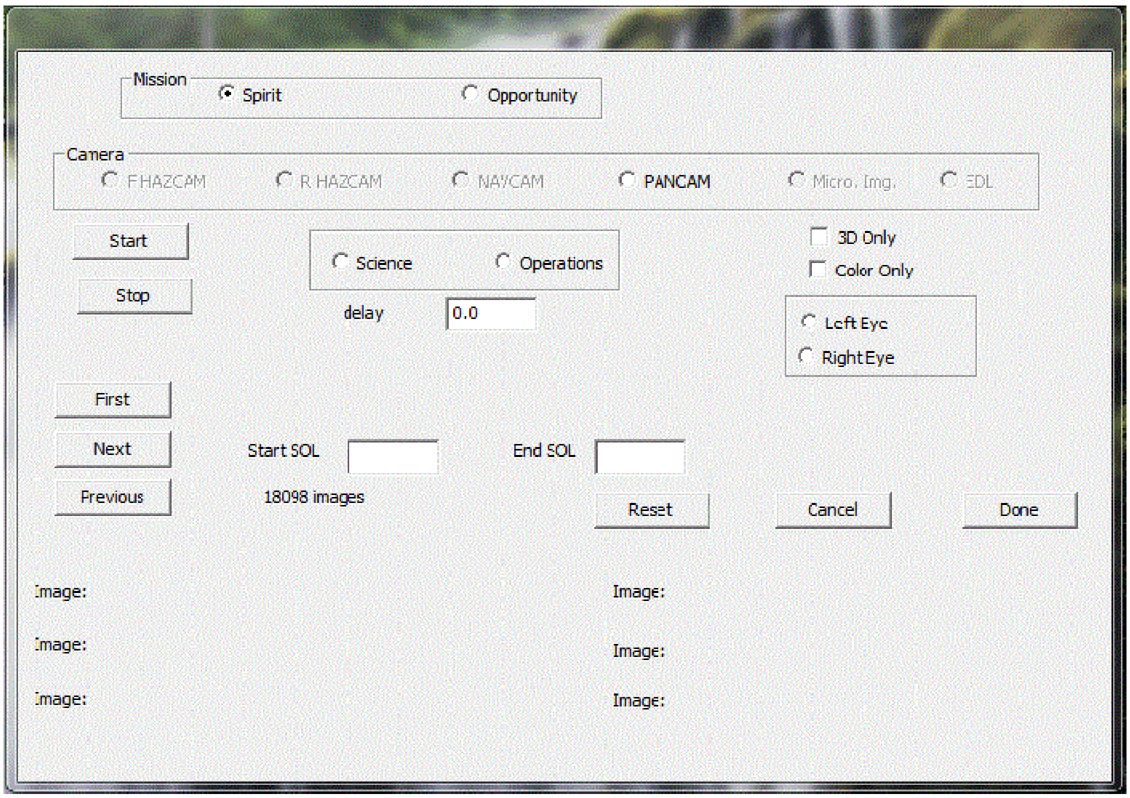

Figure 26 Mars Rover Image Selection Dialog

The information in this box is not queried from the display processes, it is derived from the file names present on the controller computer. The files are all indexed by the time the picture was acquired. Each day on Mars is called a sol and the images are grouped by which day they were taken. The actual sol they were taken on is determined by subtracting the value of the clock when the probe touched down from the current value of the clock and then dividing by the number of seconds in a Martian day. The value of 128141693 is used for opportunity's start time and 126320553 is used for Spirit's start time. The value used for the number of seconds in a Martian day is 88775.242188. The control process will attempt to locate color images, stereo images and color stereo images. The underlying assumption in finding these types of images is that each group will have the same timestamp. The control process will build a data base of all the valid filenames in the directories the user selects. All the file names will have 31 characters and the control process will discard all the files that don't have "EFF" as the product type. The control process will build a linked list of all the file names found and each node of the linked list will have all the fields parsed from the file names. Each node will also have a pointer to a chain of images for color images as well as stereo images. Each node will also have a pointer into a linked list of all the directories so that a full path to the file is retained without needing to retain the same directory path string in each node. As each file name is loaded into this linked list the entire list is always maintained sorted by the clock time from the file name. The linked list of files will drop any file that appears to be a lesser version of an existing file. This determination is done by looking at the version number contained in the file name. If the file names have only different version numbers only the highest version number is kept. There is a separate list for each of the two rovers. The user will then be able to select portions of the files that are in the database. The database is scanned to find color images, stereo images and color stereo images. Because the list is sorted by clock time, images with the same clock time will always occur near each other. Color images are only available from the Pancam and will be a set of 3 images from the left eye or two images from the right eye. The left eye color image is composed from filters three or four for the red channel, filter five for the green channel and filter six or seven for the blue channel. The right eye image is composed from filter one for the blue channel and filter two for the red channel. The green channel is not present for the right eye image. If the left eye image has both filters three and four available, the image from filter four is chosen for the red channel. If the left eye image has both filters six and seven available then the blue channel is chosen from filter six. The color images are assumed to share the same site, position and sequence number. For every unique combination of site, position and sequence number if a full set of color images exist (red, blue and green for the left eye, red and blue for the right eye) that set is assumed to be single color image. The software currently will not attempt to find a set of images which should be stitched together to form a panoramic image. Once a full set of images is found there is a color pointer set in each of the file nodes in the master linked list. This color pointer will form a linked circle of nodes for that color image. The color pointers for the left eye have the red channel pointing to the blue channel, the blue channel pointing to the green channel, and the green channel pointing to the red channel. Because the right eye's color image has only two channels, they point to each other. The control process will find stereo images in a similar manner. Stereo images are assumed to have been taken with the same time stamp in both the right and left eye images. Each node in the linked list has a stereo pointer so the right and left images will be able to point to each other.

Tools

Zoom and Pan

The user can select the current zoom level from the control process menu system. The zoom level is automatically reset each time the control process detects a change of image. The zoom level is accomplished by ordering the OpenGL to alter its zoom function. The zoom level is retained by the display processes and is primarily used to determine how still images are placed on the screen. The centering tools described later use the current zoom level to determine the image placement on the screen. The user is able to control the zoom level through the dialog box as displayed in Figure 27.

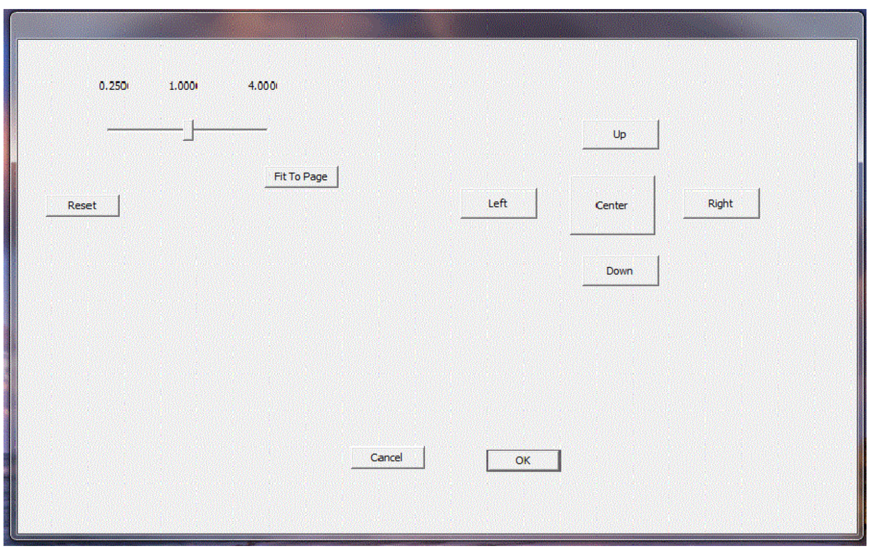

Figure 27 Image Zoom Control Dialog

The zoom slider has 5 positions in it so that to the left of the center position each of the two positions will cut the current zoom level in half. Each position to the right of the center position will double the current zoom level. This dialog box also allows the user to pan the image. The image can never be panned fully off the screen. The image will never be offset beyond the final 10% of the screen's width or height. The button marked CENTER will send a Recenter command to the display process so that the image is centered while preserving the current zoom level. The button marked Fit to Page may perform a change in the zoom level as well as centering the image using the new zoom level. The pan direction buttons have a cumulative effect. Each click of the button will push the image one pixel in one of the directions. The control process doesn't keep track of where the image has been panned to, it will continue to send Pan commands to the display process even if the display process has determined that the panning limit has been reached in that direction.

Redraw

The control process has a menu item which will order the display processes to redraw their screen. This command is only done through user interaction with the control process, this code is not executed in response to a Windows message requesting a redraw of the screen. Windows may pass the redraw message to the display programs but the code in those programs will allow Windows to redraw the screens. This could happen if the code was running in a windowed (not full screen) mode and the display window was moved. The full redrawing process is detailed elsewhere because this is really just a function of drawing the screen.

Recenter

This menu option will have the display the current image centered at the current zoom level. If the current image is larger than the current screen dimensions the center of the image will be displayed. The size of the current image depends on what the current zoom level is. The number of pixels in the current image is multiplied by the zoom level in order to determine the current image's dimensions. The zoom level is usually thought of as a single number, but the zoom level can be set to different values in the horizontal and vertical directions.

Histogram

This menu option will display a histogram of the current image. This option is controlled by the use of the dialog box seen in Figure 28.

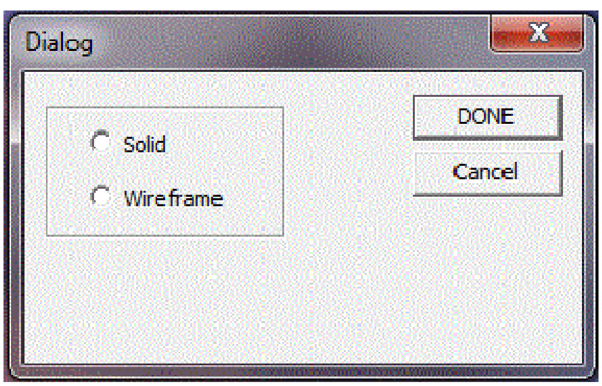

Figure 28 Histogram Control Dialog

The histogram can be displayed with either a solid view, much like a bar graph, or as a line, much like a line graph. The histogram displayed is for each of the three colors (red, green, blue). This is true for all images, even images which are in grayscale. If the user has drawn a bounding box around a part of the image the histogram will be of only the part within the bounding box. The histogram is drawn completely within the display process because the controller computer never opens the image files. The three color histograms are drawn independently of each other. The display process will determine the lowest value and highest value for each color. This will determine the number of bins that the will be needed. Because the low and high values are determined for each color, the low and high values will probably be different for each color. The highest count of values for each color is also calculated so the height of the histogram can be determined. Because the range calculations are done independently for each color the histogram will always fill the screen. The values cannot be compared between the colors.

Crop

This menu item will allow the user to select a portion of an image for display or for manipulation by the other tools. The user must first map out the area to be retained with the mouse. When the crop option is selected, the current image will become the new crop. The current zoom level is not changed. The location of the cropped image is also not changed. This has the visual effect of turning the removed pixels black, but the new image is a smaller image and the pixels that were cropped out do not factor into any of the other tools.

Swap

Images such as anaglyph images and JPS images have a separate right and left eye image embedded in them. The swap menu option will allow the user to switch the default processing of these type images by switching the right and left designation. This menu option will display a check mark next to it when selected. The selection will be retained even if a new image is loaded. This would allow the designation to be made assuming that the glasses that are being used have the default designations for right and left swapped. When this option is selected it only affects how images are processed. For example, without this feature selected the left eye display process will display the image for the left eye. When this option is selected, the display processes do not swap their right and left designations. The left eye would display the right eye image when this option is selected.

8 bit

The image viewer will convert an image file into an internal bitmapped form while preserving as much of the original resolution as possible. This means that if the original data is stored as 3 pixels (red, green, blue) each having 32 bits then the internalized form of that image file will retain that pixel's bit depth. This menu item will allow the user to rewrite the internalized form so that each pixel is one byte long. The routine will not destroy the original internalized bitmap but will create a new lower resolution bitmap. This bitmap will be built by determining the highest pixel value across all three channels and subtracting the lowest pixel value across all three channels then dividing that result by 256 and using that division factor to shrink the pixel value for each channel of each pixel. If the original bitmap is already in eight bit format, the display processes will not generate any type of error nor will they make any changes to the image data.

Mouse Movement

The mouse on the control computer can be used to construct a bounding box on the display computers. When the user pushes down the left mouse button the current mouse position is sent to the display processes and from then on every mouse movement is sent to the display computers. The display computers will construct a rectangle from where the mouse currently is to where the mouse was when the left button was clicked. This is the behavior as long as the user holds down the left mouse button. When the user releases the left mouse button that is the end of the box and the display program will no longer receive updates on the mouse position. If the user clicks the right mouse button, the current bounding box operation is cancelled. If the left mouse button is still being held down, it is ignored. The user must release and repress the left button to start a new bounding box. There is only one bounding box possible and it is not possible to edit an existing box. If the user hits F12 the control process will have the display processes turn on their cursors and will report every movement of the controller's mouse to the display devices. The F12 key is a toggle and hitting it again will turn this feature off.

Minimize/Restore

The control process can send the display programs an order to minimize their display windows. The display programs will then request Windows to minimize the current window. The actual outcome of this operation is dependent on the operating system. The display process does not make any alteration in its processing loops. The display process does not check to see if its window has been minimized before attempting to output graphics therefore a minimized window must be restored in order to be visible, the restoration will not occur as a side effect of displaying a new image. The restore operation reverses the effect of the minimize operation. This operation does not check to see if the window has actually been minimized before passing the restore command to Windows.

Echo

The echo tool is designed to measure how long it takes to send a message from the controller to the display and back. When this menu option is selected the control process will utilize Microsoft's QeryPerformanceCounter routine to measure the round trip travel time. The control process will mark the current time and then send out the echo request to the display processes. The display process will immediately send out an echo response when it receives the controller's request. The display process's echo response does not do any processing; it only sends back the response packet. When the controller receives the echo response it records the current time and will display the total travel time in a display box as seen in Figure 29.

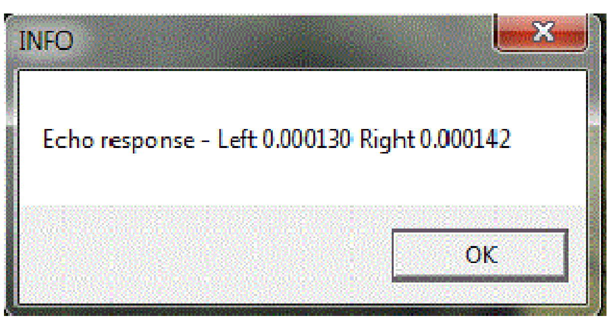

Figure 29 Round Trip Transmission Results

Asynchronous Communications

This menu item will allow the user to interact with the socket constructs used to implement communications. Each socket used can be controlled independently. The control of the sockets is accomplished by using the ioctl system routine and passing it the FIONBIO subcommand. This will turn blocking on or off. The default operation for Windows sockets is to send network messages in blocked mode, that is, as soon the write function on the socket returns that message buffer can be used by the program again because the programs knows the message has been sent. This menu item on the control computer will read “Async Comm” when blocking is in effect (allowing the user to choose to switch to non-blocking asynchronous communications) and “Sync Comm” when non-blocking mode is in effect. The control process will first toggle the left eye’s socket blocking parameter then order the display process to make the corresponding change in its socket. The control process will then repeat the operation for the right eye. If the control process is unable to toggle the setting of the blocking parameter it will not change the menu item’s entry.

Brightness/Contrast

When the user selects the brightness and contrast control from the menu the control process will open a new dialog panel as shown in Figure 30.

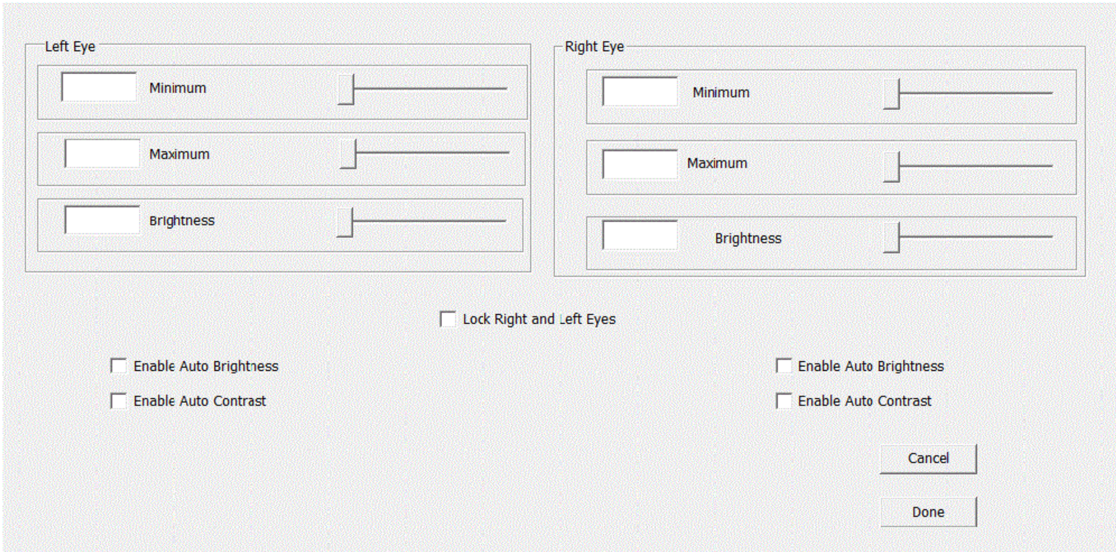

Figure 30 Brightness and Contrast Dialog

Convolution

This menu option will perform edge detection on the current image and display the results. The edge detection is done by applying the sobel operator to each point in the image.

Viewing Frustum Editor

The user can use the dialog box in Figure 31 to directly edit the viewing transformation the program is currently using. The dialog box is displayed using the current transformation information. The program initializes with a two dimensional orthographic projection. If the user wishes to manipulate the three dimensional projection it will be necessary to have a scene displayed in 3D stereo. The values entered are not checked for validity. When the update is ordered the frustum values are changed and the current image is redisplayed.

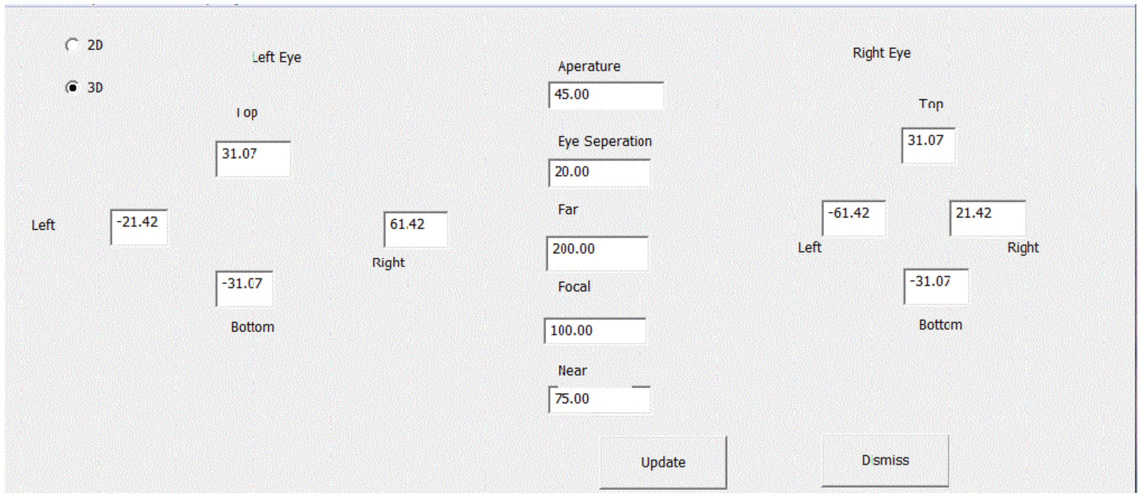

Figure 31 Viewing Frustum Editor

Model View Editor

The dialog box in Figure 32 will allow the user to alter the viewing parameters for the currently displayed model. The current screen is always redrawn using the new parameters anytime the user selects the update button. The user can enter values for a translation, rotation or scale. The values entered are not additive, each time an update is done the graphics system is reset before the update is applied. Scale factors of zero are not allowed, and any such values are quietly converted to 1.0. If the user has locked the axes scales, the scale factor on all the axes is set to the lowest of the axes. The rotation column has a button to enable the geometry to be spun about an axis. When this button is pressed, a complete rotation (360 degrees) is done in one degree increments. The inversion box at the top of this column indicates that the rotation occurs about all the other axes, excluding the axis where the spin button was pressed.

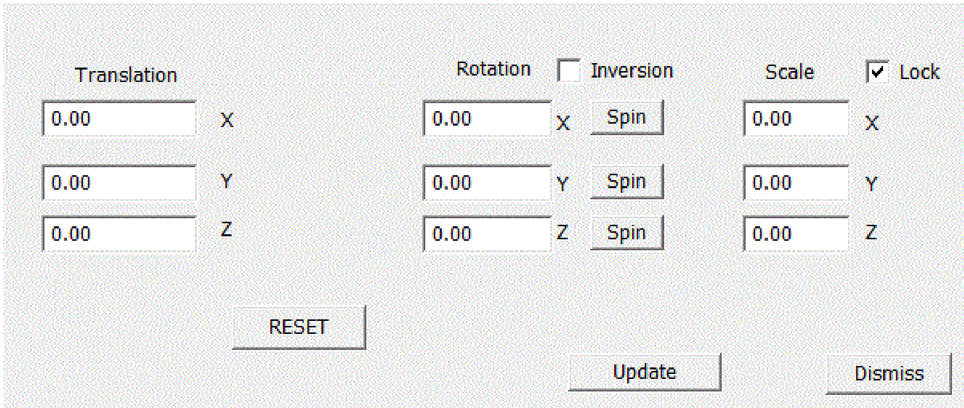

Figure 32 Model View Editor

Data Display

The visualization system is able to display data sets using parallel coordinates as well as Cartesian coordinates. There are currently three modules for reading in data sets and displaying them. The first module is for reading data sets in crystal vision format. This format has the data in text files with header information. The first line of this data file has the number of variables. The control process will skip over any blank lines as well as any blanks at the top of the file. The crystal vision files use the word 'variables' followed by a ':' with the number of variables after that. The control process will look for the first three characters of the word 'variables' on the first line and if that exists the number of variables will follow the word 'variables' and either a colon, blank or tab character. If there isn't a variables keyword on the first line the control process will look for the first line to contain only the number of variables. The second line of the crystal vision format has the word 'labels' followed by a colon. The labels then follow after this line. Each label is one to a line and there must be the same number of labels as variables. The control process does not require the use of the line with the keyword ‘labels’. However, the first label is not allowed to contain a colon. The control process does not check to see if the first line uses the word ‘labels’, it only checks to see if that line has a colon on it. If that line does contain a colon, then it is assumed to be the ‘labels’ line. The data then follows the labels and each line has a quantity of numbers equal to the number of variables. The data file can also be stored as a file of values separated by commas. If the first line of that file contains non-numeric data that line is taken to be a header line and contains the labels for the variables. The fields of the header line still need to be separated by commas. Once the header line is taken care of the data fields in the crystal vision formatted file and the csv file are processed the same. The files can be processed the same if each line of the csv file has its separator character changed from a comma to a blank. The data values that are read in must be numeric values. If the data values consist of string data the strings must be represented as numeric data externally to the visualization system. Because it is unknown how many data points will exist the data points are stored in a linked list. Each node of the linked list has one data point for each dimension, the screen coordinates of each data point and a color for each data point. This data structure is shown in Figure 33.

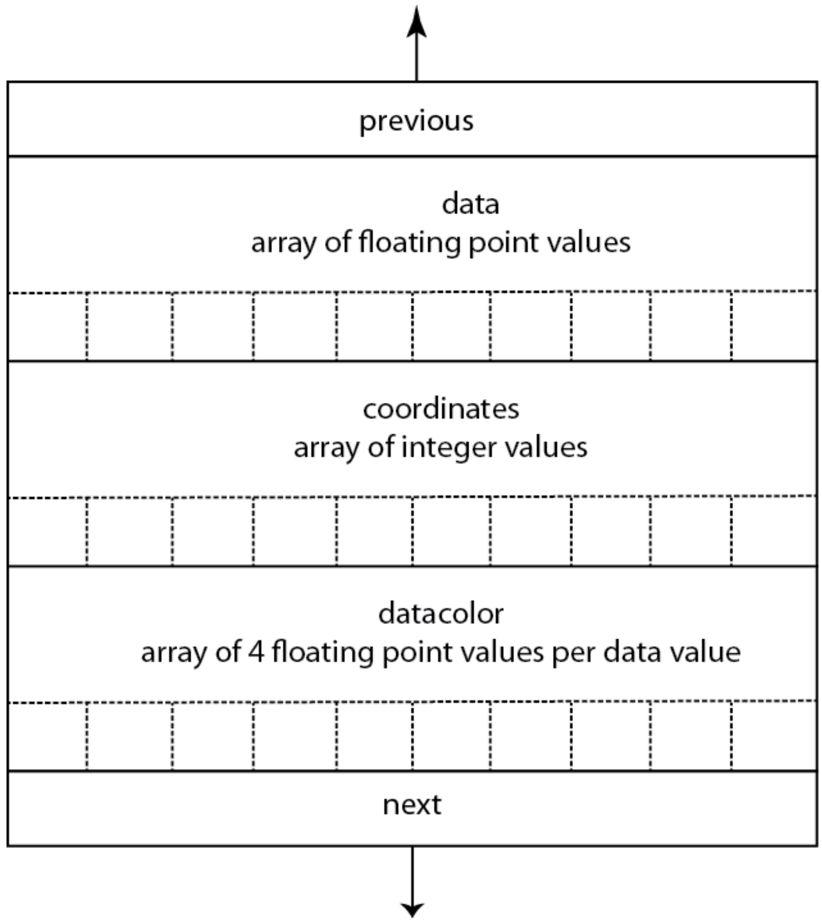

Figure 33 Data Points Data Structure

Once the display processes have parsed the data set the appearance of the axes are set up. The default arrangement of the parallel axes system will evenly distribute horizontal parallel axes down the entire screen. There is an option which will allow the axes to be drawn vertically. If there are less than four axes the top and bottom 25 pixels of the screen are not used for horizontal axes. The display processes create a separate data structure for the location and color of the axes. The data structure for the axes is show in Figure 34.

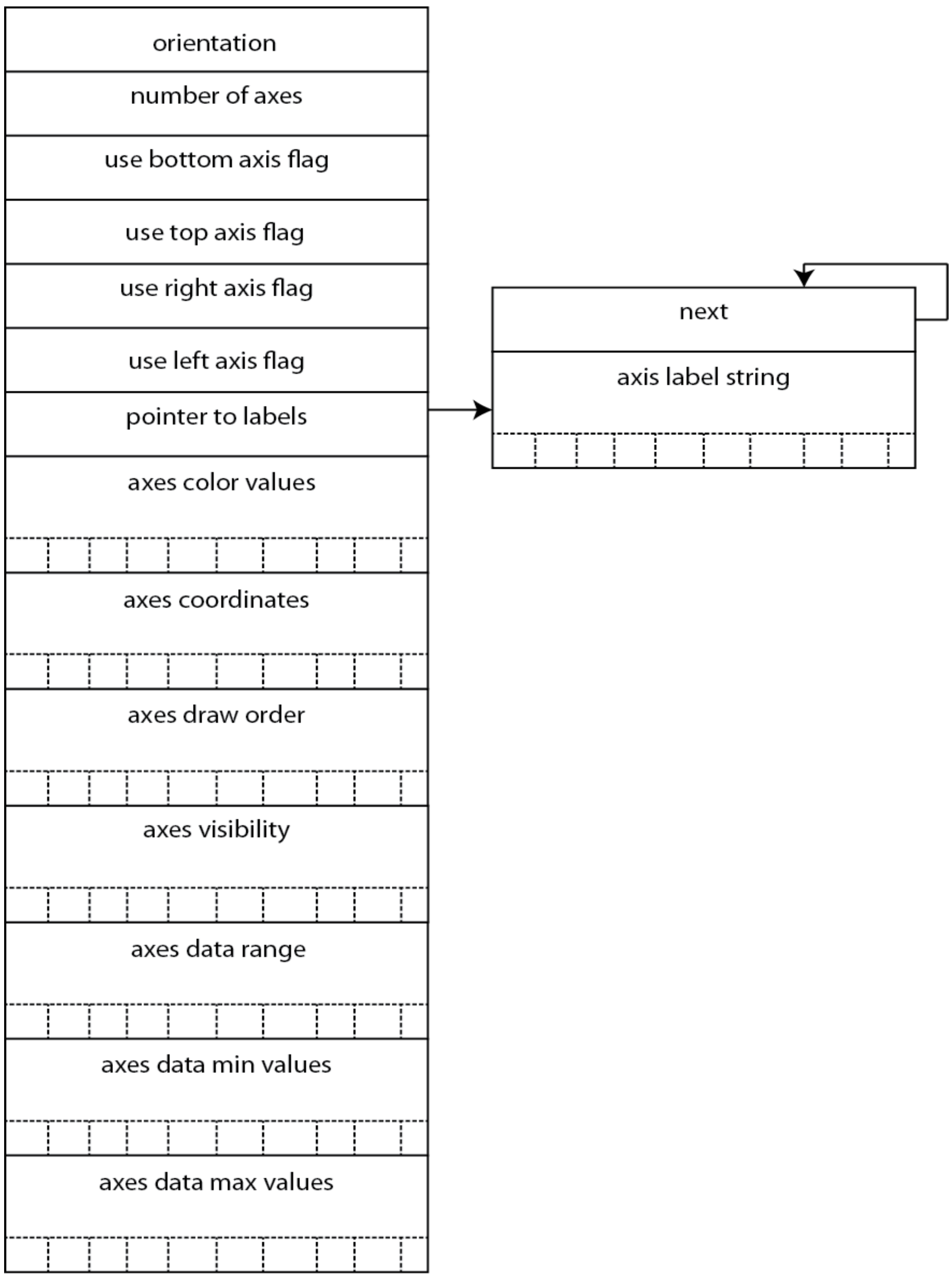

Figure 34 Axes Data structure

The display processes also maintain an integer array to control the order that the axes are shown on the screen. The control process allows the user to interact with the parallel coordinates display using with dialog box as seen in Figure 35.

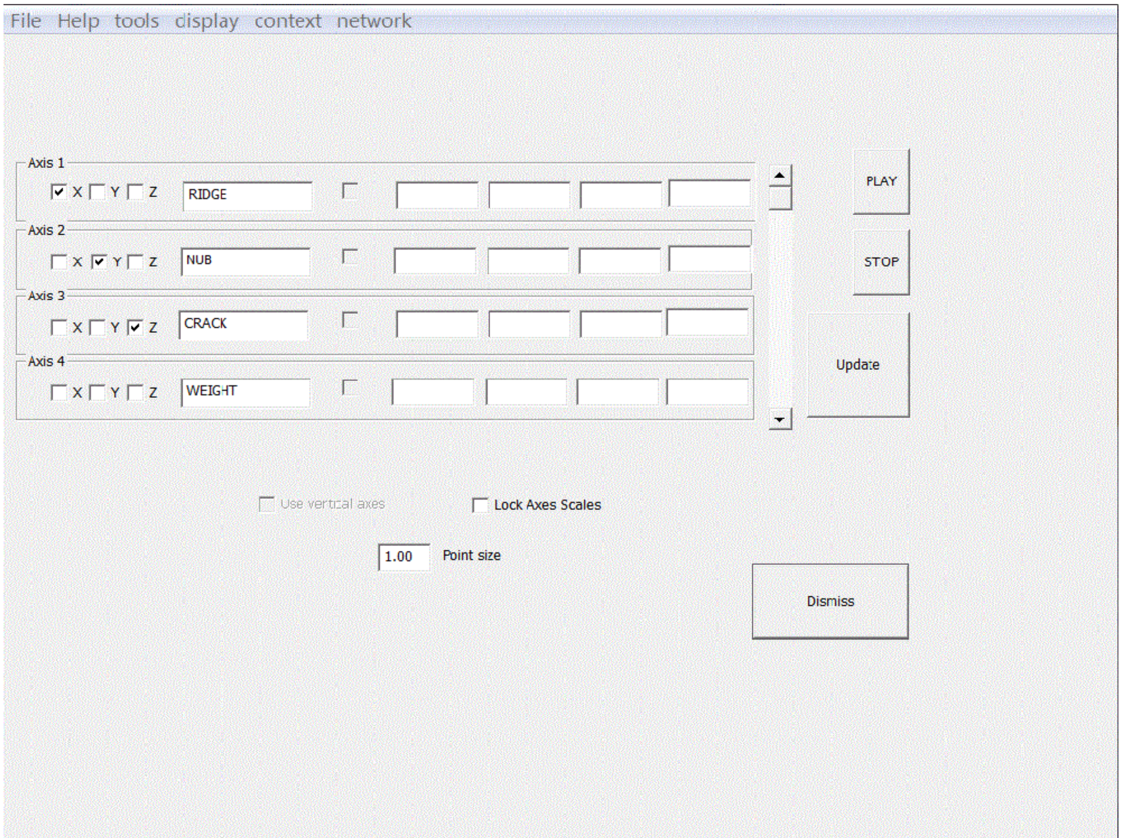

Figure 35 Axes Control Dialog

This dialog box will list the axes in the order they are displayed. The first three boxes mark the parallel coordinates mapping to Cartesian coordinates. The next box on each line will display the label and allow the user to change the label. The checkbox is to indicate visibility of the axis, or rather the check mark is to indicate that an axis is no longer visible. The next 4 edit boxes are for coloring the data on that axis, not for coloring the axis itself. There is a check box at the bottom to alter the axes' scales. When this box is checked, all the axes will share a common scale, otherwise each axis has its own scale independent of the other axes. Any changes made to these fields update the control process' data only, but those changes are not passed to the display processes until the update button is pressed. This will allow the user to perform a series of changes without partial updates being made until all the changes have been set up. The scroll bar on this dialog box will allow the user to see all possible axes If there are fewer than four axes this scroll bar is not active. The final two buttons,' play' and 'stop', allow the user to display all the possible permutations of the axes ordering. When the user selects the play option the control process will start a new processing loop where it first sends the display process a request to update to the next axis ordering and wait for all the display processes to complete that update. The control process will then check to see if there are any Windows messages that need to be handled. The control process will handle at least some of these messages before completing that iteration of the loop. The control process will not process all of the messages because Windows could post a large number of redraw requests and that could prevent the user from ever getting control of the control process execution. Because the user could click on the play button while a play operation is in progress the play button is disabled once it is pressed and will remain disabled until the stop button is pressed or the window is initialized again. When the stop button is pressed the control process will query the display processes for the current axis ordering. Once the axis ordering is received the dialog box is updated to reflect the new axis ordering.

The order of the axes is not updated while the display processes are displaying the different orderings. The control process has a separate menu item to move to the next ordering of the axes. Each time this option is selected the display processes will display the next sequential ordering of the axes. There is also an option in the menu to reset the ordering of the axes to its original ordering. Because there is an option in the menus to change the axes ordering when the axes control box dialog is started the display processes need to be queried to determine the axis ordering which is currently displayed.

Network data

The visualization system can ingest network traffic data. The data is from a series of captures conducted on campus. The entire packet is not available, but only the header information. The data files are stored in PCAP format so the visualization can ingest any network data stored in this format. The network information is stored in a linked list. PCAP format consists of a file header followed by a series of records preceded by a small header. Each record of the capture file is one packet sniffed off the network. The record header data from the capture file is converted to a data structure with four different components as seen in Figure 36.

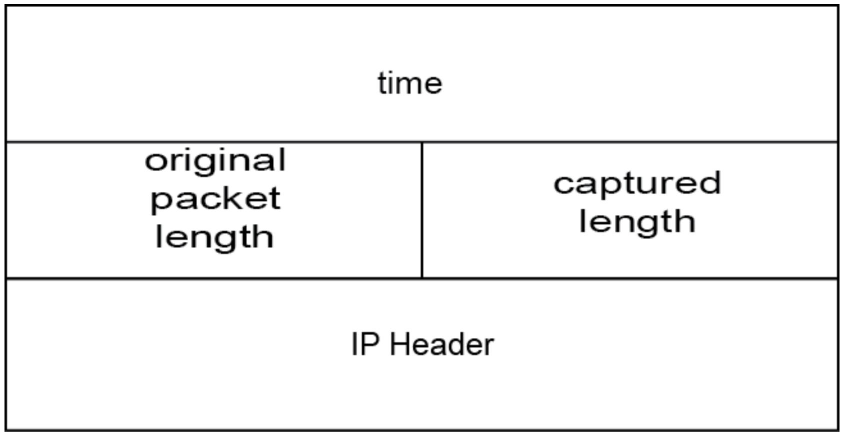

Figure 36 Network Data File Data Structure

There is a time field followed by two length fields. The two length fields represent the actual packet length as was reported to the network and the second length field is the number of bytes that were captured by the network sniffer. The IP header information is taken from the network packet information in the PCAP file. The information in the IP header is reduced to the data structure seen in Figure 37. The next part of the data structure is a block of data that is specific to the protocol that the packet was sent under. The protocol information is available in the IP header section. The data structure has a pointer to the block of data and the interpretation of that data block is dependent on translating the protocol field of the IP header. The protocol block will have information for packets sent using TCP, UDP or ICMP. Any network packets found that are not one of these three protocols are discarded and will not appear in any further analysis. The three protocol blocks are seen in Figure 38, Figure 39, and Figure 40.

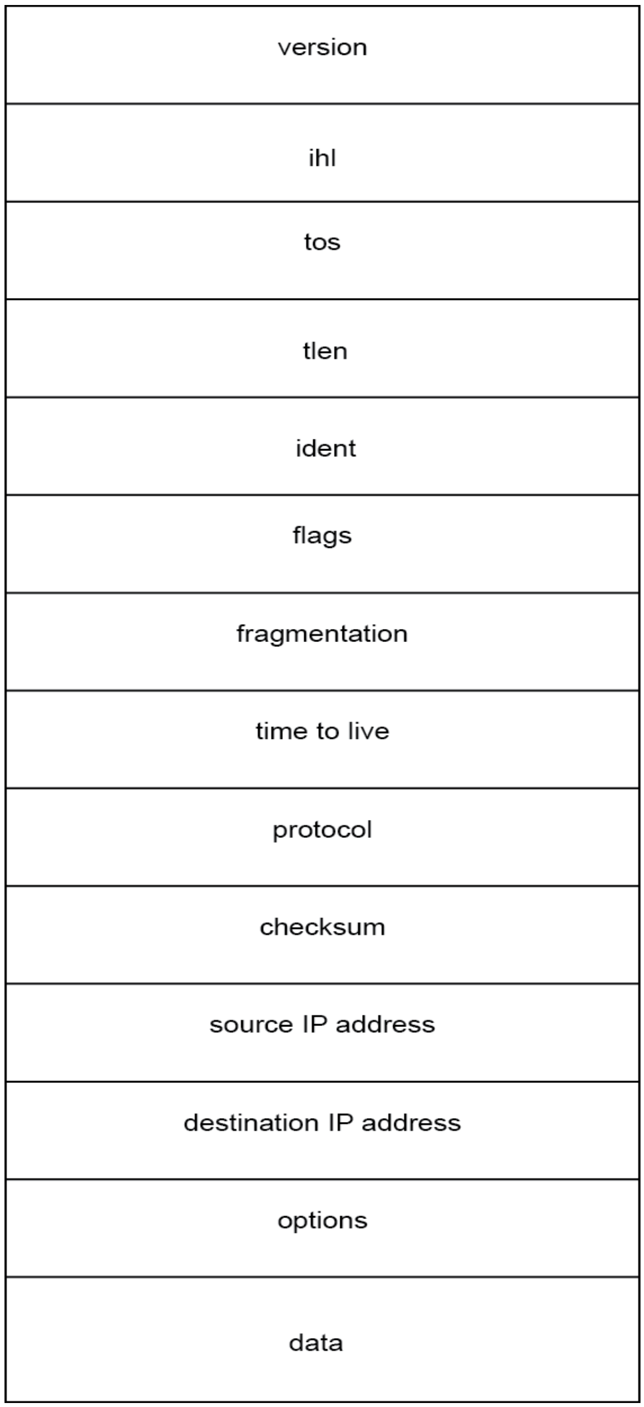

Figure 37 Network Packet Data Structure

Figure 38 TCP Packet Data Structure

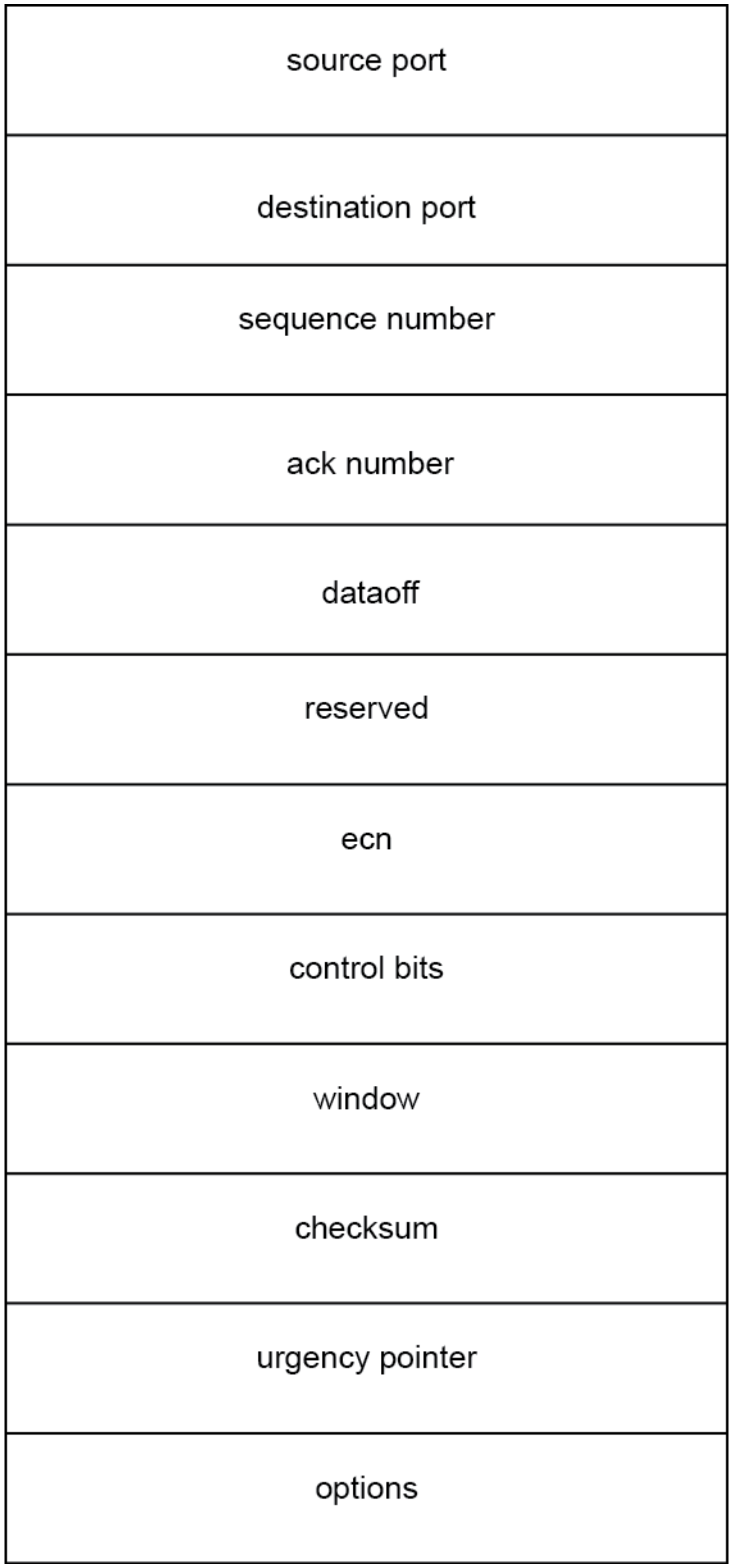

Figure 38 TCP Packet Data Structure

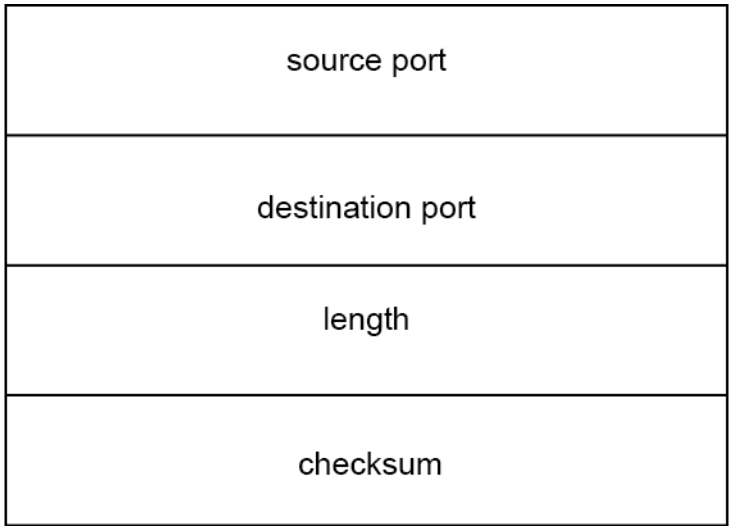

Figure 39 UDP Packet Data Structure

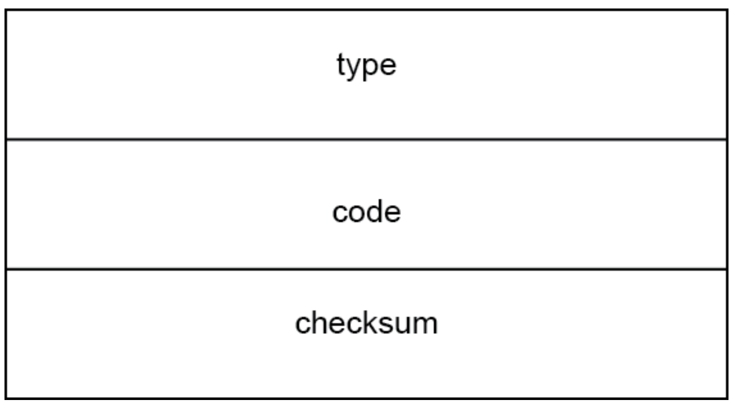

Figure 40 ICMP Packet Data Structure

The final part of the linked list is a data field which will consist of data from the packet transmission. This field is kept primarily for future expansion because the majority of our data consists of network captures that removed the most if not all of the data field.

Lottery data

The visualization system can ingest data exported from the Virginia Lottery web site. The data comes from the lottery web site saved as a file and then converted to text. The data can be parsed from the following games: Pick 3, Pick 4, Cash 5, Win for Life and Mega Millions. The lottery data set is present to show a real world random data set. The Pick 3 and Pick 4 games utilize either three or four numbers drawn from the set of zero through nine. The cash 5 game selects five numbers in the range of one through thirty-four. The Win for Life game selects seven numbers in the range of one through forty-two. The mega millions game currently selects five numbers in the range of one through fifty-six and a sixth number in the range of one through forty-six. The game has always had the format of choosing five numbers and an additional sixth number but the range that those numbers have been chosen from over the years has changed. The game began by selecting five numbers in the range of one through fifty and the sixth in the range of one through twenty-five.

Geometry Display

The display processes need to be able to respond to a request to redraw the screen at any time therefore the software must maintain the state of what needs to be drawn. The software does not maintain this state by constructing a bitmap of what should be drawn and then having that bitmap ready to display. The display processes maintain enough of the state to be able to draw the screen at any time and use this routine to render the screen for initial drawing as well as redraws. The drawing routine will either be drawing images or data. If an image is being drawn, the software maintains what is considered the current image. This image may be superseded by another image in which case the first image is discarded and the new image is now the current image. The current image may not be discarded if the new image is somehow related to the current image. For example, when a histogram is displayed the histogram will be displayed (the histogram becomes the current image) but the image that the histogram is based on is not discarded. The drawing routine has three major types of display functions, the histogram, data display and image display. The data drawing section will draw the axes based on the predetermined locations. The number of axes drawn will depend on the number of visible axes. The linked list of data points will then be used to draw every data point. The order that the data points are drawn in have a separate array so an order rearrangement will only result in a very small number of changes not an entire rewrite of the data arrays. When the drawing routine draws an image it will always transfer the contents of the current image's data array into a new buffer for display. This will facilitate displaying only a section of the image. If the entire image is to be displayed then it will still be necessary to copy from the current image into another buffer. The display routine does not attempt to reread the current frame buffer. While there are OpenGL routines which could be used to read back the current contents of the frame buffer the display routine relies on the original data to display information to the user.

CHAPTER FIVE

Conclusion

This dissertation provides an overview and a detailed implementation of a stereoscopic data and image visualization system which I developed using multiple computers as opposed to a single computer with multiple video outputs. The software system has four major components, a communications system, a display system, an image viewer and a data viewer. The following is a list of the highlights of the system I developed:

- Communications subsystem
    - Use network communication to allow each eye to have a dedicated computer system for display purposes and have a separate computer system dedicated to user interaction
    - Create a protocol to display information requests from the controller computer on the display computers
    - Allow the controller computer to request parameter information from the display computers
    - Allow the controller computer to indicate to the user (via cursor) when the display computers are processing requests
    - Measure the amount of time to pass messages between computers

- Display subsystem
    - Display three dimensional scenes using passive stereo
    - Display traditional computer graphics imagery using the same technology used for displaying stereo images
    - Display histogram information for images
    - Display the results of software edge detection on image files
    - Display three dimensional objects in stereo
    - Arbitrarily manipulate the location of three dimensional objects by scaling, rotation, and translation
    - Arbitrarily change the internal viewing parameters of the three dimensional viewing transformation (two dimensional as well)
- Image Viewer
    - Display stereo imagery by using an image file on each of two computers, one computer for the right eye and one computer for the left eye
    - Display stereo images by using the same image file on each computer but showing only half of the image to each of the right and left eye computers
    - Display stereo images by using a separate file for each of the red, green and blue channels for each eye
    - Display animated stereo and non-stereo imagery

    - Display DSLR images acquired with a stereo slide bar by specifying the amount of eye separation
    - Display imagery from the NASA Mars Rover mission by user configurable parameters
    - Display MRI slice images
    - Display mammography images
    - Display full color images from the Visible Human Project
    - Display CT slice images
    - Display any portion of an image
    - Locate the displayed image to an arbitrary position on the screen
- Data Visualization subsystem
    - Display data in parallel coordinates
    - Show all possible orderings of the axes in parallel coordinates
    - Change the color of all the data points on any axis in parallel coordinates
    - Remove an arbitrary axis from a parallel coordinates display
    - Display data as a two dimensional scatter plot
    - Display data as a three dimensional stereo plot
    - Display data files in crystal vision format
    - Display data files in a CSV format
    - Display data from the Virginia lottery website
    - Display data from PCAP network traffic captures

Because this system relies on multiple computers a handheld device such as a tablet or phone could be used to control the input to the system. The visualization system can be expanded to include much of the functionality of GGobi, or could be used as a backend display system for GGobi.

Future Work

This visualization system could be expanded to provide multiple nodes for each of the right and left eyes. The most difficult part of this undertaking would be determining where the projectors for each eye overlap. This could be done by projecting a solid gray image to each projector. If we assume that the projector and the screen being projected onto can provide a relatively evenly illuminated image then a digital camera could examine the resulting output of the projects and determine if there is a brighter section in the middle where the projectors have overlapping images. The two computers for that eye could begin to decrease the intensity in columns of pixels until the overlapping columns have the same intensity as the rest of the image. Because the visualization system generally builds an internal bitmap of images to be projected the intensities of the pixels in the overlapping columns would be trivial to reduce.

The current visualization system is built on the idea of a single stereoscopic display system. The system could be expanded to include traditional LCD display panels in conjunction with a stereoscopic display system. These LCD panels could be used to provide supplemental material about the visualization being seen on the stereoscopic display. The most difficult part of implementing this would be to design a user interface panel that a novice user could use to direct output to the appropriate display without presenting so many combinations of displays that the panel is overwhelming with options. In addition, it would be useful to provide a scripting language so that data and images could be presented without operator intervention. This scripting language would have to allow each event to last for a specific duration or to run until it completes before moving on to the next step in the script.

The controller program could be moved to a tablet device. This tablet would allow the user to control the visualization system wirelessly and interact with the system via touch controls. Because the images to be displayed reside on the display nodes as well as the controller computer it would only be necessary to store thumbnail sized representations of the images to be displayed on the tablet. The tablet would also be able to provide the user with a voice input system to execute most commands.

REFERENCES

[1] J. Symanzik, "Three-Dimensional Stereoscopic Plots," *Wiley Interdisciplinary Reviews: Computational Statistics,* vol. 3, no. Nov-Dec, pp. 483-496, 2011.

[2] C. Wheatstone, "On Some Remarkable, and Hitherto Unobserved, Phenomena of Binocular Vision. Part One," *Philosophical Transactions,* vol. 128, pp. 371-394, 1838.

[3] New York Public Library, "Stereoscopic Photography," [Online]. Available: http://stereo.nypl.org/about/stereoscopy. [Accessed 19 April 2012].

[4] D. Phillips, "A Brief Stereoscopic History," [Online]. Available: http://home.centurytel.net/s3dcor/history.htm. [Accessed 19 April 2012].

[5] A. Woods, "The Illustrated 3D Movie List," [Online]. Available: http://www.3dmovielist.com/list.html. [Accessed 21 April 2012].

[6] E. J. Wegman and Y. Said, "Color Theory and Design," *Wiley Interdisciplinary Reviews: Computational Statistics,* vol. 3, no. 2, pp. 104-117, 2011.

[7] J. C. d'Almeida, "Novel Appareil Stereoscopique {in French}," *Comptes Rebdus des Seances l'Academie Sci*, vol. XLVII, pp. 61-63.

[8] E. Wegman and D. DePriest, Statistical Image Processing and Graphics, New York: Marcel-Dekker Press, 1986.

[9] E. Wegman and Q. Luo, "On Methods of Computer Graphics for Visualizing Densities," *Journal of Computational Graphical Statistics*, vol. 11, no. 1, pp. 137-162, 2002.

[10] C. Cruz-Neira, D. J. Sandin and T. F. DeFanti, "Surround Screen Projection Based Virtual Reality: The Design and Implementation of the CAVE," in *SIGGRAPH'93: Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, 1993.

[11] E. Wegman and J. Symanzik, "Immersive Projection Technologies for Visual Data Mining," *Journal of Computational Graphical Statistics,* vol. 11, no. 1, pp. 163-188, 2002.

[12] wikipedia, "Silver Screen," [Online]. Available: http://en.wikipedia.org/wiki/Silver_screen. [Accessed 19 March 2012].

[13] netblender.com, March 24, 2011.

[14] J. T. Fulton. [Online]. Available: http://neuronresearch.net/vision/pdf/standeye.pdf. [Accessed 20 March 2012].

[15] Wikipedia, "Stereo Realist," [Online]. Available: http://http://en.wikipedia.org/wiki/Stereo_Realist. [Accessed 19 March 2012].

[16] geowall.org, [Online]. Available: http://www.geowall.org/. [Accessed 19 March 2012].

[17] pokescope.com, [Online]. Available: http://www.pokescope.com/. [Accessed 19 March 2012].

[18] R. Burdick, "Viewer - Stereo Pair Viewer," [Online]. Available: http://http://www-users.cs.umn.edu/~wburdick/geowall/viewer.html.

[19] esri.com, "ArcGIS 3D Analyst," [Online]. Available: http://www.esri.com/software/arcgis/extensions/3danalyst/index.html. [Accessed 19 March 2012].

[20] simwright.com, "StereoGIS," [Online]. Available: http://http://www.simwright.com/stereogis.htm. [Accessed 19 March 2012].

[21] Z-Flux, "Stereo Rendering of Real-time, Interactive Animations," [Online]. Available: http://http://www.z-flux.net/. [Accessed 19 March 2012].

[22] ggobi.org, "GGobi Interactiva and dynamic graphics," [Online]. Available: http://www.ggobi.org/. [Accessed 19 March 2012].

[23] D. Swayne, D. Cook, A. Buja, D. T. Lang, H. Wickham and M. Lawrence, "GGobi Manual," [Online]. Available: http://www.ggobi.org/docs/manual.pdf. [Accessed 19 March 2012].

[24] E. J. Wegman, "Voice-controlled immersive virtual reality system". Patent 6448965, 2002.

[25] Cooperative Institute for Meteorological Satellite Studies (CIMSS), University of Wisconsin – Madison, USA., "Stereoscopic views of Hurricane Bill," [Online]. Available: http://cimss.ssec.wisc.edu/goes/blog/archives/date/2009/08/20. [Accessed 19 March 2012].

[26] NASA, "Overview," [Online]. Available: http://marsrover.nasa.gov/overview/. [Accessed 20 March 2012].

[27] Neon Helium, "Neon Helium Productions lessons 01-05," [Online]. Available: http://http://nehe.gamedev.net/tutorial/lessons_01__05/. [Accessed 19 March 2012].

[28] P. Bourke, "3D Stereo Rendering Using OpenGL (and GLUT)," [Online]. Available: http://paulbourke.net/miscellaneous/stereographics/stereorender/. [Accessed 19 March 2012].

[29] Microsoft, "glFrustum function," 6 March 2012. [Online]. Available: http://msdn.microsoft.com/en-us/library/dd373537(v=vs.85).aspx. [Accessed 22 April 2012].

[30] D. A. Clunnie, "Stanford PVRG JPEG Patches for GE DICOM MOD," [Online]. Available: http://www.dclunie.com/jpegge.html. [Accessed 20 March 2012].

[31] NASA, "Summary," [Online]. Available: http://marsrover.nasa.gov/gallery/edr_filename_key.html. [Accessed 19 March 2012].

[32] glprogramming.com, "The Official Guide to Learning OpenGL, Version 1.1," [Online]. Available: http://www.glprogramming.com/red/. [Accessed 20 March 2012].

[33] K. McDonnell and K. Mueller, "Illustrative Parallel Coordinates," in *Eurographics/ IEEE-VGTC Symposium on Visualization 2008*, 2008.

[34] J. Johansson, P. Ljung, M. Jern and M. Cooper, "Revealing Structure within Clustered Parallel Coordinates Displays," in *IEEE Symposium on Information Visualization*, 2005.

[35] A. Laine, "Digitally Enhanced Mammography," [Online]. Available: http://www.fathom.com/feature/60812/. [Accessed 21 March 2012].

[36] H. Aydinoglu and M. H. I. Hayes, "Stereo Image Coding: A Projection Approach," *IEEE Transactions on Image Processing,* vol. 7, no. 4, pp. 506-516, 1998.

[37] T. Duff and G. J. Ledin, "OpenGL Lectures glFrustum Case Studies," [Online]. Available: http://www.cs.sonoma.edu/other/CS375_OpenGL_Slides_pdf/Perspective_gluFrustum.pdf. [Accessed 19 March 2012].

[38] P. Hsu and Y. Tseng, "Visualization of Hyperspectral Images for Spectral Analysis," *Journal of Photogrammetry and remote Sensing,* vol. 9, no. 3, pp. 81-96, 2004.

[39] B. P. Lelieveldta, M. Sonka, L. Bolinger, T. D. Scholz, H. Kayser, R. van der Geest and J. H. Reiber, "Anatomical Modeling with Fuzzy Implicit Surface Templates: Application to Automated Localization of the Heart and Lungs in Thoracic MR Volumes," *Computer Vision and Image Understanding,* vol. 80, no. 1, pp. 1-20, 2000.

[40] H. Zhou, X. Yuan, H. Qu, W. Cui and B. Chen, "Visual Clustering in Parallel Coordinates," in *Eurographics/IEEE-VGTC Symposium on Visualization*, 2008.

[41] E. J. Wegman and Y. Said, "Natural Homogeneous Coordinates," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. Nov-Dec, pp. 678-685, 2010.

[42] Z. Cho, J. Jones and M. Singh, Foundations of Medical Imaging, New York: John Wiley and Sons, 1993.

[43] A. Kak and M. Slaney, Principles of Computerized Tomographic Imaging, New York: IEEE Press, 1988.

[44] N. Thompson, 3D Graphics Programming for Windows 95, Redmond, Washington: Microsoft Press, 1996.

[45] A. Unwin, M. Theus and H. Hofmann, Graphics of Large Datasets: Visualizing a Million, New York: Springer, 2006.

[46] W. W. J. Orrison, J. D. Lewine, J. A. Sanders and M. F. Hartshorne, Functional Brain Imaging, St. Louis: Mosby, 1995.

[47] W. Rollman, "Zwei neue stereoskopische Methoden," *Annals of Physical Chemistry*, vol. 90, pp. 186-187, 1853.

[48] W. T. Hewitt, S. Larkin and A. J. Grant, "Techniques for Visualizing Multidimensional Data," in *20th CERN School of Computing*, Pruhonice, Czech Republic, 1997.

[49] S. Sun and J. Chen, "The Application of Full-Scale 3D Anthropometric Digital Model System On Breast Reconstruction of Plastic Surgeries," *Biomedical Engineering-Applications, Basis & Communications*, vol. 15, no. 5, pp. 200-206, 2003.

[50] J. N. Maki, J. F. I. Bell, K. E. Herkenhoff, S. W. Squyres, A. Kiely, M. Klimesh, M. Schwochert, T. Litwin, R. Willson, A. Johnson, M. Maimone, E. Baumgartner, A. Collins, M. Wadsworth, S. T. Elliot, A. Dingizian, D. Brown, E. C. Hagerott, L. Scherr, R. Deen, D. Alexander and J. Lorre, "Mars Exploration Rover Engineering Cameras," *Journal of Geophysical Research*, vol. 108, no. E12, 2003.

[51] S. P. Harbison and J. G. L. Steele, C - A Reference Manual, Englewood Cliffs, NJ: Prentice Hall, 1991.

[52] A. Buja, D. Cook, D. Asimov and C. Hurley, "Theory of Dynamic Projections in High-Dimensional Data Visualization," [Online]. Available: http://stat.wharton.upenn.edu/~buja/PAPERS/paper-dyn-proj-math.pdf. [Accessed 20 March 2012].

[53] E. J. Wegman, "Hyperdimensional Data Analysis Using Parallel Coordinates," *Journal of the American Statistical Association*, vol. 85, no. 411, pp. 664-675, 1990.

[54] C. Wheatstone, "On Some Remarkable, and Hitherto Unobserved, Phenomena of Binocular Vision. Part Two," *Abstracts of the Papers Communicated to the Royal Society of London*, vol. 6, pp. 138-141, 1852.

[55] M. Tory, S. Potts and T. Moller, "A Parallel Coordinates Style Interface for Exploratory Volume Visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 2005, no. Jan-Feb, pp. 71-80, 2005.

[56] C. Vertan, B. Ionescu and M. Ciuc, "Color Image Edge Detection by Parallel Coordinates-Based Color Distance," in *8th International Conference on Communications*, 2010.

[57] W. M. Wells, P. Viola, H. Atsumi, S. Nakajima and R. Kikinis, "Multi-modal Volume Registration by Maximization of Mutual Information," *Medical Image Analysis,* vol. 1, no. 1, pp. 35-51, 1996.

[58] G. Tritto, M. C. Tritto and G. Pirlo, "The Analytical Descriptors of Three-Dimensional Geometry of Human Organs: Application to the Analysis of the Female Breast Shape," in *IEEE Engineering in Medicine & Biology Society 11th Annual International Conference*, 1989.

[59] G. W. Braudaway and F. Mintzer, "Techniques for Locating the Stereo Axis of Stereo-Pair Images," in *1991 International Conference on Acoustics, Speech, and Signal Processing*, Toronto, Ont. , Canada, 1991.

[60] H. Takahashi and F. Tomita, "Large Scale Stereo Vision," in *11th IAPR International Conference on Pattern Recognition, 1992. Vol.I. Conference A: Computer Vision and Applications, Proceedings.*, The Hague , Netherlands, 1992.

[61] J. W. Hill, P. S. Green, J. F. Jensen, Y. Gorfu and A. S. Shah, "Telepresence Surgery Demonstration System," in *1994 IEEE International Conference on Robotics and Automation, 1994. Proceedings.*, 1994.

[62] S. M. Wiener and S. M. Dunn, "Automatically Registering Stereo Pairs of Tomographic Imagery," in *Proceedings of the 1993 IEEE Nineteenth Annual Northeast Bioengineering Conference, 1993.*, 1993.

[63] D. V. Papadimitriou and T. J. Dennis, "Epipolar Line estimation and Rectification for Stereo Image Pairs," *IEEE Transactions on Image Processing,* vol. 5, no. 4, pp. 672-676, 1996.

[64] S. Mitra, R. Long, S. Pemmaraju, R. Muyshondt and G. Thoma, "Color Image Coding Using Wavelet Pyramid Coders," in *Proceedings of the IEEE Southwest Symposium on Image Analysis and Interpretation, 1996.*, 1996.

[65] J. Jiang, E. A. Edirisinghe and H. Schroder, "Algorithm for Compression of Stereo Image Pairs," *Electronics Letters,* vol. 33, no. 12, pp. 1034-1035, 1997.

[66] F. Aghdasi, R. K. Ward and B. Palcic, "Noise Filtering for Mammographic Images," in *14th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 1992*, 1992.

[67] Y. Kita, R. Highnam and M. Brady, "Correspondence Between Different View Breast X-Rays Using a Simulation of Breast Deformation," in *1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1998. Proceedings.*, Santa Barbara, CA, 1998.

[68] S. Ahn, "OpenGL Projection Matrix," [Online]. Available: http://www.songho.ca/opengl/gl_projectionmatrix.html. [Accessed 20 March 2012].

[69] R. Highnam, M. Brady and B. Shepstone, "A Representation for Mammographic Image Processing," *Medical Image Analysis*, vol. 1, no. 1, pp. 1-18, 1996.

[70] G. Szekely, A. Kelemen, C. Brechbuhler and G. Gerig, "Segmentation of 2-D and 3-D Objects from MRI Volume Data Using Constrained Elastic Deformations of Flexible Fourier Contour and Surface Models," *Medical Image Analysis,* vol. 1, no. 1, pp. 19-34, 1996.

[71] J. F. Brinkley, S. W. Bradley, J. W. Sundsten and C. Rosse, "The Digital Anatomist Information System and Its Use in the Generation and Delivery of Web-Based Anatomy Atlases," *Computers and Biomedical Research*, vol. 30, pp. 472-503, 1997.

[72] N. Ruiter, R. Stotzka and T.-O. Muller, "Model-Based Registration of X-Ray Mammograms and MR Images of the Female Breast," *IEEE Transactions on Nuclear Science*, vol. 53, no. 1, pp. 204-211, 2006.

[73] E. J. Wegman and D. B. Carr, "Statistical Graphics and Visualization," in *Handbook of Statistics 9: Computational Statistics*, C. R. Rao, Ed., Amsterdam: North Holland, 1993, pp. 857-958.

CURRICULUM VITAE

Roger Duane King graduated from James Madison High School, Vienna, Virginia, in 1980. He received his Bachelor of Science from George Mason University in 1986. He has been employed as an engineer at George Mason University since 1987 and received his Master of Science in Computer Science from George Mason University in 1992.