5-11-2002

Using Virtual Environments to Visualize Atmospheric Data: Can It Improve a Meteorologist'S Potential to Analyze the Information?

Sean Bernard Ziegeler

Recommended Citation

Ziegeler, Sean Bernard, "Using Virtual Environments to Visualize Atmospheric Data: Can It Improve a Meteorologist'S Potential to Analyze the Information?" (2002). *Theses and Dissertations*. 4917.
https://scholarsjunction.msstate.edu/td/4917

USING VIRTUAL ENVIRONMENTS TO VISUALIZE ATMOSPHERIC DATA: CAN IT IMPROVE A METEOROLOGIST'S POTENTIAL TO ANALYZE THE INFORMATION?

By

Sean Bernard Ziegeler

A Thesis
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in Computer Engineering
in the Department of Electrical and Computer Engineering

Mississippi State, Mississippi

May 2002

USING VIRTUAL ENVIRONMENTS TO VISUALIZE ATMOSPHERIC DATA: CAN IT IMPROVE A METEOROLOGIST'S POTENTIAL TO ANALYZE THE INFORMATION?

By

Sean Bernard Ziegeler

Approved:

James C. Harden
Graduate Coordinator of the
Department of Computer Engineering

Robert J. Moorhead
Professor of Computer Engineering
(Director of Thesis)

Lori M. Bruce
Assistant Professor of
Electrical Engineering
(Committee Member)

Joerg Meyer
Assistant Professor of
Computer Science
(Committee Member)

A. Wayne Bennett
Dean of Engineering

Name: Sean Bernard Ziegeler

Date of Degree: May 11, 2002

Institution: Mississippi State University

Major Field: Computer Engineering

Major Professor: Dr. Robert J. Moorhead

Title of Study: USING VIRTUAL ENVIRONMENTS TO VISUALIZE
ATMOSPHERIC DATA: CAN IT IMPROVE A METEOROLOGIST'S
POTENTIAL TO ANALYZE THE INFORMATION?

Pages in Study: 60

Candidate for Degree of Master of Science

Conventional analysis of atmospheric data includes three-dimensional desktop-computer displays. One disadvantage is that it can reduce the ability to zoom in and see small-scale features while concurrently viewing other faraway features. This research intends to determine if using virtual environments to examine atmospheric data can improve a meteorologist's ability to analyze the given information.

In addition to possibly enhancing small-scale analysis, virtual environments technology offers an array of possible improvements. Presented is the theory on developing an experiment to establish the extent to which virtual environments assist meteorologists in analysis. Following is the details of an implementation of such an experiment. Based on the quantitative results obtained, the conclusion is that immersion can significantly increase the accuracy of a meteorologist's analysis of an atmospheric data set.

DEDICATION

To Ann-Marie, for her considerable moral support and common-sense advice.

ACKNOWLEDGEMENTS

The author would like to thank the members of his graduate committee, Dr. Robert Moorhead, Dr. Lori Bruce, and Dr. Joerg Meyer, for the knowledge and wisdom provided for this research and in my graduate education.

The author would like to acknowledge Dr. Paul Croft, Duanjun Lu, and Ruilian Jiang of Jackson State University, Department of Physics, Atmospheric Science, and Science Education for providing atmospheric data and meteorological insights.

This research was conducted as part of the High Performance Visualization Center Initiative (HPVCI) funded by the Department of Defense High Performance Computing Modernization Program (HPCMP).

TABLE OF CONTENTS

CHAPTER Page

APPENDIX

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Weather forecasting and severe storm prediction are just two examples of applications that depend upon the analysis of atmospheric data. If any improvements could be made to such analyses, the benefits of better weather forecasting and severe storm prediction would have significant, positive consequences. This includes a better understanding of hurricanes, tornadoes, thunderstorms, flooding and other weather conditions that have a strong impact on people's everyday lives. The goal of this research is to investigate the possibility of such improvements using a combination of scientific visualization and virtual environments. Refer to Figure 1.1.

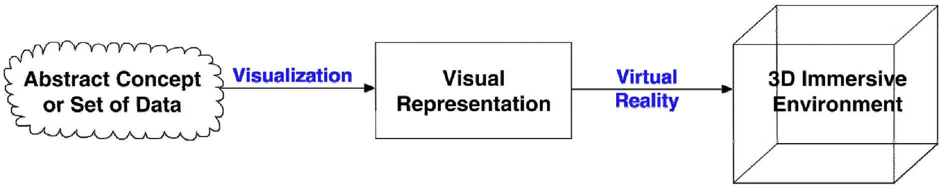

Fig. 1.1 Combining scientific visualization and virtual environments.

1.1. A Background in Scientific Visualization

Scientific visualization is the first aspect of this research. The origins of scientific visualization are deeply rooted in the past, with the purpose of using the visual sense to grasp non-visual, abstract concepts [13]. The following quote from René Descartes in

1637 summarizes this notion [9]: "Imagination or visualization and in particular the use of diagrams has a crucial part to play in scientific investigations." In the twentieth and twenty-first centuries, scientific visualization has been used to interpret the large volumes of numbers generated by computer models or observed by electronic equipment [32]. From [19], scientific visualization can be defined as follows: a tool utilizing the human sensory modalities to understand complex, abstract information and gain insight that would otherwise be unavailable. In recent years, this includes the utilization of computer graphics as a primary means of visualization.

1.2. A Background in Virtual Environments

While scientific visualization is important to this research, the second aspect, virtual environments, is the primary focus. Ivan Sutherland established the foundation of computerized virtual environments in 1968 with his research on a head-mounted display device using monitors anchored in front of the eyes and a head-tracking device [28]. Since then, other technologies have arisen, such as projection-based systems [11], various position tracking systems, and the software to manage this hardware [3, 21, 36]. A good working definition of virtual environments (for the purposes of this research) can be found in [10]: an "immersive, interactive, multi-sensory" computer-based system capable of generating a convincing alternative view of some scene or setting.

1.3. How Could Virtual Environments be Useful?

Previous research has shown that the use of virtual environments can result in a user seeming to be immersed within some sort of information display. This begs the question, is such immersion useful for scientific visualization in general? More specifically, could it be used for something like meteorological visualization? And if so, what exactly about this immersion would be useful?

The answer to the above general question lies in the literature. Research that documents utilization of virtual environments is summarized and cited. However, to gain an insight into the answer to the above specific question, experiments must be done to test the thesis that virtual environments are useful for meteorological visualization. In this thesis, the design, implementation, and results of such an experiment will be discussed.

CHAPTER II

PREVIOUS RESEARCH

There has been considerable research on the use of virtual environments in both visualization and task performance. The literature involving scientific visualization has been concerned with several disciplines, *e.g.*, geographic, outer-space, or chemical visualization [12, 25, 26]. Many of the studies involving task performance originate from the field of tele-presence and tele-robotics [1, 5, 14, 22, 24, 33, 34]. The use of virtual environments in scientific visualization will be discussed first, followed by the studies involving task performance.

2.1. Key Terms

There are two key terms used in virtual environments research that should be kept in mind. The words may seem to have a similar context, but they are in fact quite different:

- *Immersion* – the objective ability of the hardware and software to encompass the user in the virtual environment that is being presented [5, 24].
- *Presence* – the subjective feeling that the user actually believes that he or she is “inside” of the virtual environment [5, 14, 24, 33].

2.2. Virtual Environments in Scientific Visualization

Upon inspection of the literature, three basic properties of virtual environments emerge with respect to scientific visualization: exploration, interaction and analysis. Past research can be classified according to which of these properties it investigates.

2.2.1. Exploration

Usually one does not know beforehand where the most important features are in a given large data set. Often, the most important features that may affect decisions are located in a small region of a data set. Furthermore, it is uncommon for the researchers involved to know exactly where to find these features. Exploration is the only way to find them, and virtual environments are the most effective means of exploration [23].

Virtual environments and scientific visualization work well together [4]. Visualization does not attempt to realistically represent abstract concepts; instead, it attempts to convey the information in the most effective means possible. Virtual environments provide an explorative interface for information, even if a representation of the information does not exist in reality. At this point, the fusion of scientific visualization and virtual environments becomes evident: the visualization can produce a non-realistic, but accurate, representation and the virtual environments can effectively explore the representation.

According to [26], visualization usually requires one or both of two perspectives. The first is a “big picture” overview. The second, an in-depth view of the details within the data, is probably more suitable for virtual environments. According to the authors, virtual environments offer natural methods for such exploration of details. Users can immerse themselves in regions of data to more closely investigate. The result is an intuitive exploration of space and time.

2.2.2. Interaction

Virtual environments provide an intuitive interface to manage the visualization [4]. In [25], the authors summarize a project involving virtual environments and molecular visualization. The display is superior because the users are immersed within and surrounded by molecules. The users can directly and intuitively interact with the individual molecules.

Finally, the authors of [26] make a very important observation about virtual environments: users can explore the data as if in the real world. There is no need to rely on the user's ability to learn, understand, and operate a complex interface.

2.2.3. Analysis

With respect to analysis, virtual environments offer three advantages. Someone can:

- Gather a quick overview of the data without making the display too busy or confusing.
- Inspect specific areas of the data without excess effort.
- Compare features in different locations without obscuring the display [4].

In the precursor to this research [38], the authors listed several advantages to using virtual environments as opposed to standard desktop workstations. They were gathered from two sources and are evaluated as follows:

- Improved human-computer interaction for navigation and inspection [37].
- A "visual paradigm" more closely related to the data [10].
- Stereoscopy for improved depth perception [37].
- Wider field-of-view to see more data simultaneously [37].

The authors also established an important additional advantage – virtual environments can assist with analyzing small-scale features within the data [38]. The rationalization for this is that one can zoom closely into an area without losing the surrounding information or cluttering the display. Also, with the wide field-of-view, one can still see the surrounding information. This hypothesis was demonstrated qualitatively using evaluations from cooperating meteorologists.

In all of the above research, one theme seems to endure: while desktop visualization and virtual environments are both acceptable for overview visualization, virtual environments seem superior for analyzing small-scale features. In addition, the natural interface of virtual environment technology allows for better exploration of the data in order to locate features.

2.3. Research on the Effects of Immersion

Most research indicates that immersion can have a significant effect on task performance, including in scientific visualization. There is some general research into this area, but most research focuses on some specific component of immersion to see if it affects task performance.

2.3.1. Initial General Research

The authors of [5] state the level of immersion can simultaneously affect the user's sense of presence and task performance ability. This is further reinforced by [33], which finds that performance-increasing variables include stereoscopy, resolution, and other immersion-related variables. The primary reason that the research so strongly supports immersion's effect upon performance is that immersion is such an easily definable quantity. Thus, objective experiments can be developed to test the relationship between immersion and performance [24].

2.3.2. Specific Task Oriented Studies

According to [1], human performance with teleoperated systems is often decreased because the "visual modality" is two-dimensional. To have the brain reconstruct a three-dimensional scene, two conditions are required: (1) binocular vision – the presence of two eyes with a certain distance between detecting two separate images, (2) stereoscopic vision – the ability to merge two images in the brain. In general, taking advantage of these capabilities using technology is referred to as stereoscopy. Stereoscopy is one component of immersion that can be used to improve task performance.

Stereoscopy can be especially useful for tasks in reduced visibility conditions. The authors of [22] show that users gained a 17% increase in task performance with clear visibility and a 25% increase under severely degraded visibility. Thus, not only can stereoscopy assist in general, but also if there is low visibility (e.g., cluttered display or occlusion due to semi-transparent objects), stereoscopy can be advantageous.

Stereoscopy can also make visualizations more understandable [30]. The authors of one study show that stereoscopy can make it possible to increase the amount of information in a three-dimensional graph by a factor of 1.6 and maintain the same comprehensibility.

Field-of-view usually refers to the horizontal angle of space directly in front of a person covered by a display. Research by [12] asserts that a wide field-of-view improves maneuverability. This improvement lies in the increased peripheral vision available with a wide field-of-view. The conclusion drawn here is that field-of-view is essential for exploration.

Another study on field-of-view shows increased task performance upon increasing the field of view [34]. The subjects of this experiment were asked to track targets and shoot them when they were determined to be threats. The targets would shoot at the users in a predefined amount of time. The experiment was repeated with various fields-of-view ranging from 20 to 120 degrees and with varying complexities (number of targets to simultaneously track). The results depended upon the task complexity. For the less complex tasks, 10% fewer targets were hit, and the user was vulnerable for 8% longer when using 20 degrees of field-of-view instead of 120 degrees. For more difficult tasks, 45% fewer targets were hit and the user was vulnerable for 43% longer when using

20 degrees instead of 120 degrees. Thus, field-of-view can be especially important for very complex tasks.

The authors of [30] discuss the benefits of head tracking. The use of head tracking in their study improved visualization quality. It was possible to increase the amount of information in a three-dimensional graph by a factor of 2.2 and retain the same comprehensibility (recall that stereoscopy effected a 1.6 factor of improvement). Thus head tracking can be significantly beneficial, depending upon the application.

A comprehensive study done by [6] concludes that a diverse range of immersive mechanisms can improve task performance, including stereoscopy, field-of-view, and tactile feedback. Tactile feedback refers to any information provided to the user through the sense of touch. This includes force feedback, heat generation, etc. All things considered, immersion is a powerful ally in the quest for improved task performance.

2.4. Research on the Effects of Presence

The sense of presence is essential in making a virtual environment seem realistic to the user, thus it is considered invaluable by the entertainment industry [33]. However, the usefulness of presence for task performance has had mixed reviews. In most of the literature, there is one consensus: there is no reason that increased presence should necessarily directly cause an increase in task performance [5, 14, 24, 33].

According to [33], there is generally a correlation between presence and task performance, but usually no causal connection. In fact, immersion is usually the reason for the correlation between presence and performance – when immersion increases so does presence, and so does task performance [5].

So, immersion may improve the sense of presence, but presence may not be desirable. Not all tasks require a strong sense of presence [14]. Certain tasks require an unrealistic display of information, thus presence may debilitate performance. In addition, presence may be distracting to the user [24]. Given these findings, presence is most likely ineffective for meteorological visualization in a virtual environment.

2.5. Initial Response

The design and implementation of the virtual environment software [38] is discussed in Chapters III & IV. For now, it is sufficient to say that during the implementation of the software, some cooperating meteorologists tested the application. The responses from these experts helped to guide the further development of the software and the design of future user studies.

The meteorologists responded positively, with the following observations:

- The vertical analysis was better than in two-dimensions.
- The small-scale analysis was better than traditional desktop three-dimensional visualization, especially in combination with the vertical analysis.
- The navigation was more natural in the virtual environment.
- The stereoscopy was useful in discriminating between near and distant features.
- The wider field-of-view allowed the users to see more data simultaneously.

CHAPTER III

THEORY

The foundation for this research was an experiment to quantify the effects of immersion on the performance of a meteorologist in a virtual environment. This experiment must somehow compare the performance using a standard desktop computer (or something similar) versus a virtual environment.

3.1. Meteorological Visualization System

The first step necessary in establishing some sort of experiment is the acquisition or development of software to visualize the atmospheric data. This software must be able to intuitively display multi-layer, time-series data sets, thus some sort of three-dimensional visualization will be key.

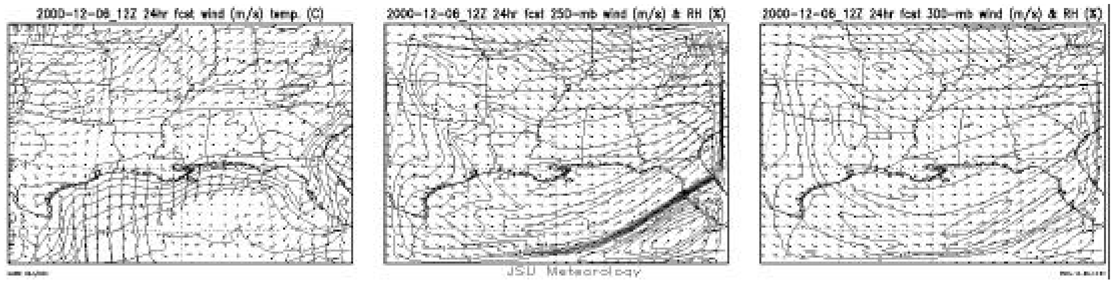

Fig. 3.1 Three layers from a forecast model data set.

Figure 3.1 shows a common two-dimensional visualization approach [38]. To gain an understanding of the vertical structure, one would have to imagine each of the images spatially superimposed on top of each other.

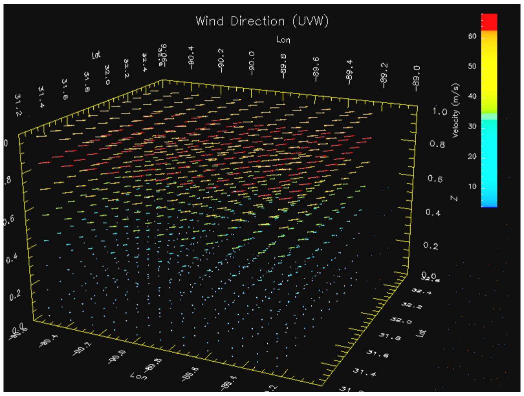

Fig. 3.2 Multiple layers from a forecast model.

In the three-dimensional example shown in Figure 3.2, one can get a better sense of the vertical structure. Vertical structure is important because of its relationship with atmospheric instability. Atmospheric instability is one factor that indicates a storm [38]. Thus we must have software capable of displaying three-dimensional information. In addition, this software must be easily transferable to virtual environment hardware.

3.1.1. Previously Existing Software

As of this publication, there are two previously existing software systems available for visualization that fit the above criteria. The first, Cave5D [7, 35], was developed as an extension of a commonly used, freeware meteorological visualization package known as Vis5D [15]. Cave5D uses the same basic computation and rendering engine, but the interface was modified to operate using virtual environment hardware.

The second software package, vGeo, [29] is designed for broad-spectrum use in geosciences. Cave5D and vGeo use the CAVELibs programming libraries [21] to interface with virtual environment hardware.

Unfortunately, both Cave5D and vGeo have some shortcomings that would make it difficult to work with our data sets. For example, neither program handles large data sets well. They assume that memory is an inexhaustible resource and attempt to store all data in some form within memory. When the data sets are large enough, this reliance on the operating system to handle memory paging can cause poor performance, especially if the organization of the data causes irregular disk accesses [27]. The recently released Cave5D 2.0 [8] may have solved some of these problems, but at the time of the beginning of this study, it had not yet been released. Finally, the CAVELibs must be purchased to use Cave5D or vGeo, and vGeo must be purchased itself.

3.1.2. Software Development

The alternative is to develop custom software for the purposes of this experiment. This is probably the superior solution for this case, since it would be more suitable for the given data sets.

One requirement for the software is a flexible but high-performance data format. The data format needs to be flexible to allow for data of various formats. It must also allow for high-performance reading because the visualizations will demand high-speed rendering, which will require fast delivery of the data. In addition, the data format must allow for multiple time steps and multiple variables since most atmospheric data is time series and consists of many variables. Once again, the format must maintain efficient management of this time-series multivariate data because of the performance requirements.

The software must offer practical visualization tools that can display different types of data effectively, yet maintain acceptable graphics performance. Atmospheric data can consist of many types of variables including wind, temperature, pressure, water concentrations, etc. Tools need to be available that can properly represent such information. However, for interactive visualization, frame rates should be at least 15 frames per second [20], thus the rendering will need to be fast.

So the user can properly interpret the information, the software should provide sufficient annotations and legends to accompany the visualization tools. This includes color bars indicating references between colors and values, political boundaries, point locations (e.g., cities), terrain height fields, etc.

Finally, a solid, object-oriented design is required for flexibility. A good design will allow for easy additions and modifications as the implementation progresses. This will also be especially beneficial when integrating the software with a virtual environment toolkit.

3.2. Virtual Environment Systems

The second step in establishing an experiment is to have our visualization software operate within some sort of virtual environment apparatus. This accomplishment requires some sort of virtual environment hardware platform and a software toolkit for managing the hardware.

3.2.1. Virtual Environment Software

A flexible, multi-platform toolkit or library is desirable for the virtual environment software. A number of platforms may be required for the proper conduction of an experiment. The toolkit must allow for the construction of the various necessary visualization tools, and not significantly reduce the performance of the application.

There are several options for virtual environment toolkits. The CAVELibs were developed for handling projection-screen based virtual environment hardware [21]. In addition, they offer an easy to use simulator mode that operates using a standard desktop-oriented computer. The CAVELibs provide a set of functions that can be called from a program to initialize and use the virtual environment hardware. They are C-based, function-oriented, mature libraries. This library-oriented architecture allows for flexible and fast applications.

VR Juggler is a more general, platform-independent set of libraries [3]. Like the CAVELibs, it interfaces with any of the projection-based hardware. However, it can work with other types of virtual environment hardware such as head-mounted displays. Similarly to the CAVELibs, there is a simulator mode for standard desktop workstations. VR Juggler is a set of C++ classes that must be derived to provide the required functionality. Thus it is an object-oriented library, so it will produce flexible, fast applications, and it allows for a well-organized software design. It is also open-source software.

Sense8’s WorldToolKit is a similar platform-independent library [36]. It supports a range of hardware and provides a C or C++ interface so that object-oriented designs are an option. It can also be configured to work on a standard desktop workstation. However, its driver for CAVE-like devices (a type of projection-based system) is no longer supported. Its ports to UNIX platforms do not include many of the tools available in its Win32 versions. Its design is oriented toward head-mounted displays driven from PC's. WorldToolKit is a powerful, mature, commercially available development library.

3.2.2. Virtual Environment Hardware

Given the toolkits described above and their simulator modes, there is no absolute need for special virtual environment hardware. However, providing the components of immersion presented in Chapter II (i.e., stereoscopy, field-of-view, head tracking, etc.) is necessary.

There are several options for an experiment. The first option is to start with a standard desktop workstation and add stereoscopic and head tracking equipment. However, this will do little to improve the field-of-view. Second, we may use some sort of head-mounted display. However, these can be bulky and significantly strain a user. Finally, some sort of projection-based system can be used, but these are large and take up considerable space.

It may be possible to use a combination of the above systems; however, this is probably not a good idea. The experiments should be run with the same system every time, minimizing the number of superfluous variables that could corrupt the results. This almost immediately eliminates the possibility of a hybrid desktop system. However, a head-mounted display or projection system is still a possibility.

3.3. Focus on Quantifiable Components of Immersion

As discussed in Chapter II, presence is difficult to quantify and isolate from other variables. In addition, there is no evidence to show that presence may be useful for meteorology. However, previous research has shown that immersion may have something to offer for atmospheric visualization. Thus, the experiment should focus on studying the effects of immersion.

The three most commonly researched components of immersion (see Chapter II) with respect to task performance and geosciences are: head tracking, stereoscopy, and field-of-view. To form a quantitative experiment, each of these must be tested individually and all possible combinations should be tested. This would result in eight total studies. This is unrealistic because we will want users to perform significant tasks that may take a considerable amount of time. If we chose two of the three components, we could reduce the total number of studies to four.

Based on the literature, especially from the results of [38], the best choices are stereoscopy and field-of-view. They should have the most significant effects on atmospheric analysis. There are no direct effects of head tracking that can be found to influence meteorological task performance.

3.4. Observation of User Studies

With two specific components of immersion, we need to establish a well-defined experimental procedure. The experiment should be a user study which evaluates user performance in all of the following combinations of stereoscopy and field-of-view. See Table 3.1.

A	No stereoscopy, narrow field-of-view
B	Stereoscopy, narrow field-of-view
C	No stereoscopy, wide field-of-view
D	Stereoscopy, wide field-of-view

Table 3.1 The four combinations of stereoscopy and field-of-view.

Thus, the results would yield the effects of stereoscopy, field-of-view, and a combination of the two on the users' performances.

3.4.1. Experimental Requirements

Most objective immersion studies involve a set of tasks for users to complete within a regulated virtual environment [1, 12, 22, 30, 31, 33, 34]. In this case, the regulated virtual environment would be a data set or group of data sets. The users would be given a set of tasks to complete in each of four data sets or in the same data set.

Since the users will be performing one set of tasks per combination of immersion components (see Table 3.1 above), the exact same data set should not be used. This would amplify the learning effects; for example, a user would begin to know this data set very well and his/her performance would improve dramatically simply because the user knows where all of the information is located.

Nevertheless, to prevent too many differences between each set of tasks, the data sets should use the same geographical location, and a different set of time steps, but with similar atmospheric conditions. Using the same geographical location helps prevent the data sets from being too dissimilar in difficulty. But if the time steps are different (*i.e.*

different days) then the user will not become too familiar with the data sets. Finally, the atmospheric conditions of each data set should be similar to prevent difficulty differences. In other words, we are attempting to strike a balance between learning effects and task difficulty variations.

3.4.2. Sets of Tasks

The sets of tasks should put various meteorological abilities to the test. In [38], the effects of immersion on certain abilities are explored. These abilities include the following:

- The ability to locate and analyze small-scale features.
- The ability to locate and analyze wide-ranging features.
- The ability to simultaneously compare features.
- The ability to make predictions or forecasts.

3.4.3. Training

In user studies involving task performance, one of the most troublesome factors can be learning effects [22]. Learning effects, with respect to task performance, are defined as effects that cause a user's performance to improve over time due to the user learning how to perform the tasks better given the tools at his or her disposal.

Training the users "sufficiently" will reduce the learning effects. Sufficient training does not indicate an exact quantity. However, Figure 3.3 shows how enough training will diminish the learning effects [22].

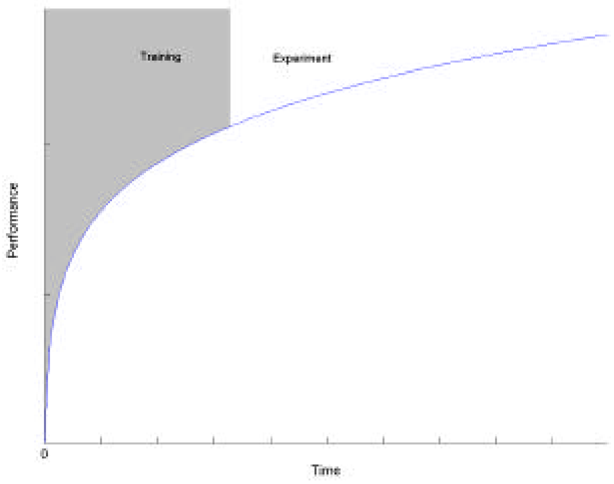

Fig. 3.3 An interpretation of the shape of a learning curve, described in [22].

Since most of the learning is done up front, a user should be trained during this initial phase. Thus, the highest increase of performance due to learning effects will occur during training. This should minimize, but not eliminate, the learning effects.

3.4.4. Control Group

Learning effects, order effects, and other factors will inevitably skew the results of the user studies. As discussed above, the learning effects cannot be completely eliminated. In addition, order effects may or may not exist in the experiment. Order effects are the result of the user completing the sets of tasks in a certain order (e.g., A, B, C, D from Table 3.1) in such a way that the performances would be different than if the tasks were completed in a different order [22].

Another problem is that some of the sets of questions may be harder than others. All of the questions should be similar in nature: they should all ask for the same types of information from analogous locations at various times. However, finding certain locations may be harder than other locations, or the data may be arranged in such a way that it is more difficult to make an analysis. Finally, other interfering effects such as exhaustion and random distractions would interfere with the results.

Given all of the above variables, it would be nearly impossible to accurately model or predict the effects. The solution is to select a reasonable percentage of the subjects to be in a control group. The control group subjects would perform all of the task sets in only one combination of stereoscopy and field-of-view. So while the standard user group performs tasks in each of the four configurations in order (i.e., A, B, C, D from Table 3.1), the control group performs the same sets of tasks in the normal virtual environment configuration (i.e., D, D, D, D). Ideally, the performance of each set of tasks should be equal, since the conditions would not have changed. Thus, the control group should reveal the accumulation of all of the interfering effects.

CHAPTER IV

IMPLEMENTATION

The following chapter describes the full implementation of the required software, illustrates the selection of hardware, and outlines the procedures necessary in a user study.

4.1. MetVR Software

The software is known as MetVR (Meteorology in Virtual Reality). There are two elements required for a usable implementation: visualization elements and virtual environment elements. Visualization elements pertain to the aspect of the software handling the visual representation of the data. Virtual environment elements pertain to the management of the virtual environment hardware, input devices, and rendering. The incorporation of these elements into the software is discussed.

4.1.1. Visualization Elements

After determining the need to develop custom software for the user studies, the first step was to determine what types of quantities would be visualized with the software. We would need tools to properly convert these quantities to visual objects.

From cooperation with meteorologists in the first year of development [38], we determined that the following quantities would be most commonly needed:

- Wind vectors
- Surface scalars (temperature, accumulated rainfall, pressure, etc.)
- Water concentrations in the form of snow, ice, vapor (clouds), rain, etc.
- Other volume scalars (temperature, pressure, etc.).

To visualize surface scalars, we have the terrain from the data set rendered as a height field. Then we allow the user to color map the terrain based on a scalar value. So, for example, the surface temperature could be mapped onto the terrain.

Wind vectors are drawn directly as vector glyphs. These are basically three-dimensional arrows, with a straight tail originating at the data point location and extending in the direction of the vector value. At the end of the tail is an arrowhead. The vectors can be uniform length or varied based on magnitude. The vectors can also be uniformly colored, color mapped by magnitude, or color mapped by vertical height.

Other volumetric scalars are rendered simply as a point field. Points placed at each *n* grid locations, where *n* is defined by the user (for subsampling). These points are color mapped by the scalar value associated with the point field. This technique is an alternative to direct volume rendering [16], which is generally too expensive for interactive frame rates.

In addition, a new visualization technique to visualize water concentrations was developed. Concentration values in meteorology are usually a mass per mass quantity (e.g., kilograms/kilogram) of water in the air. This water may be in the form of rain, snow, ice, vapor, etc. It is usually important to see several of these variables simultaneously. Our technique involves placing small particles at grid points with relatively higher values. The user may set the maximum number of particles per grid point. Based on the maximum, the number of particles is linearly interpolated from the data value. The respective numbers of particles are randomly scattered near the grid point. These particles may be associated with a glyph (e.g., a raindrop or snowflake) and may be given a color. We call this a *particle density field*. With this technique, it is possible to show multiple variables simultaneously.

Isosurfaces [18] are used for binary quantization of variables. This is especially useful to visualize clouds, which can be derived from water vapor concentrations or relative humidity or both.

Finally, many of the above tools have an associated annotation or legend. Any color-mapped object has a color bar, which indicates the mapping from color to a value. The particle density field has a "particle bar," which shows the mapping from a number of particles to a value. The terrain can be overlayed with latitude and longitude lines, political boundaries, and cities. An example exocentric view is shown in Figure 4.1.

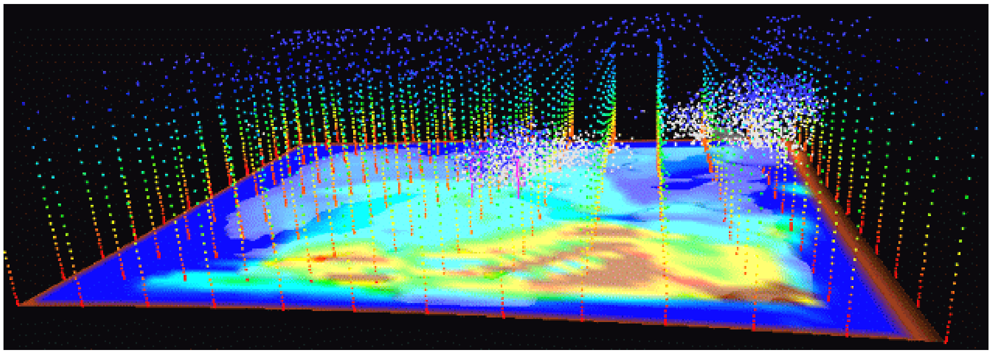

Fig. 4.1 A picture from MetVR in simulator mode.

4.1.2. Virtual Environment Elements

I chose to use VR Juggler [3] as the virtual environments library, for a number of reasons. It is open-source software, so if modifications to the library are necessary, it is possible. It is object-oriented, so I can design the software with an object-oriented methodology. See Appendix A for details on the internal design. In addition, VR Juggler is mostly platform independent, allowing for flexibility in the future. See Chapter III for details on VR Juggler and other virtual environments libraries.

The navigation controls in the virtual environment use a "flying vehicle" model [31]. There are three types, eye-in-hand, scene-in-hand, and flying vehicle. According to the author's research, exploration of a large geospatial data set is most intuitively explored as if flying within some sort of moving vehicle.

4.2. CAVE™-like Device

In Chapter III (Theory) the considerations as to which type of virtual environment system should be used were presented. A head-mounted display can be bulky and exhausting for a user. Projection-based systems can consume a large amount of space. However, since we have such a system already located on site, already taking up the required space, it was an obvious choice.

This system is a projection-based system similar to the CAVE™ (CAVE Automatic Virtual Environment) [11]. A CAVE is a set of projections screens, usually two to six that surround the user in a cubic shape. Our CAVE-like device, located in the Computerized Virtual Environment (COVE), consists of four screens: three walls and a floor. See Figure 4.2 for an artist's rendition of our CAVE-like device.

Fig. 4.2 An artist's rendition of our CAVE-like device with four screens.

The COVE is equipped with active stereoscopic shutter glasses. These glasses shutter between the left and right alternatively at 60 Hz, and the display is synchronized such that it renders a left or right eye image respectively. This is how stereoscopy is simulated. A wand can be used to interact with the virtual environment software. It has four digital buttons and two analog controls – the functions of these are configurable in software. The wand and glasses have a tracking sensor attached to each so that the position and orientation of the head and wand are tracked.

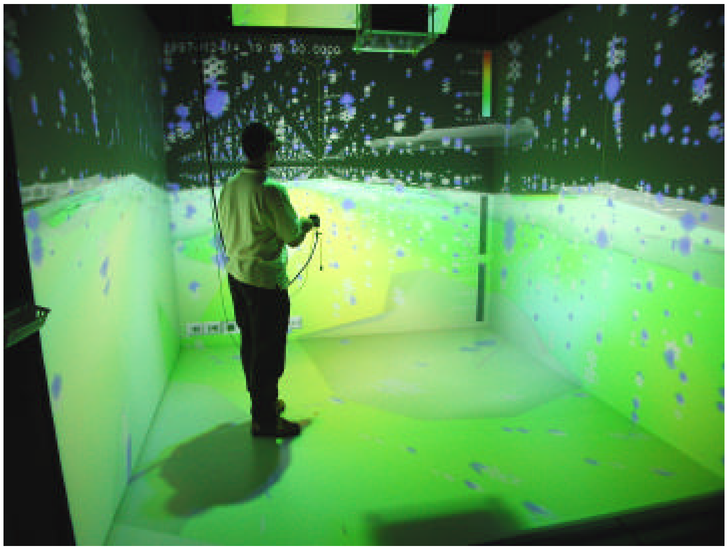

Fig. 4.3 A photograph of a user in the COVE using MetVR.

4.3. User Studies

4.3.1. Configurations of the COVE

Stereoscopy and field-of-view were chosen as the focus of the user studies. Figure 4.4 below illustrates the four possible combinations or configurations of the COVE. One set of questions is reserved for each of the configurations.

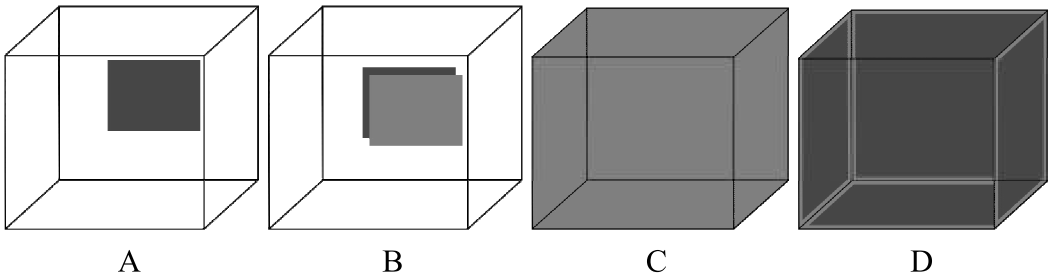

Fig. 4.4 The four possible configurations of the COVE, respectively: (A) no stereo, narrow field-of-view; (B) stereo, narrow field-of-view; (C) no stereo, wide field-of-view; (D) stereo, wide field-of-view.

4.3.2. Questions

Table 4.1 shows an example set of questions.

1	What is the surface temperature of McComb at 12:00?
2	What is the direction and velocity of wind 1000 meters above Columbus at 3:00?
3	What is the ice concentration value 8000 meters above 29.5N 86.9W at 6:00?
4	Is the total cloud concentration above Columbus higher or lower than the total concentration above 28N 87.4W at 12:00, and by how much?
5	Where and when is the lowest temperature on the surface in this data set? Give the time. Give the longitude and latitude. Give the nearest city.
6	Can you give me a surface temperature forecast for Greenville three hours after the last time step?
7	What is the sky condition forecast for Natchez three hours after the last time step (i.e. clear, partly cloudy, mostly cloudy, overcast, rain, thunderstorms, hail, and/or snow)? List all that apply.

Table 4.1 An example set of questions for a given COVE configuration.

The times and places of these questions will be different for each configuration of the COVE. As discussed in Chapter III, this will minimize learning effects. In addition, one set of questions will be reserved for a training data set. This will also assist in minimizing learning effects. See Appendix B for a list of all questions used in the study.

4.3.3. Data Sets

All of the data sets are located at the same geographic region: most of Mississippi, some of Louisiana, Alabama, Arkansas, and Florida. Mississippi is the central focus of the data set, as shown below in Figure 4.5.

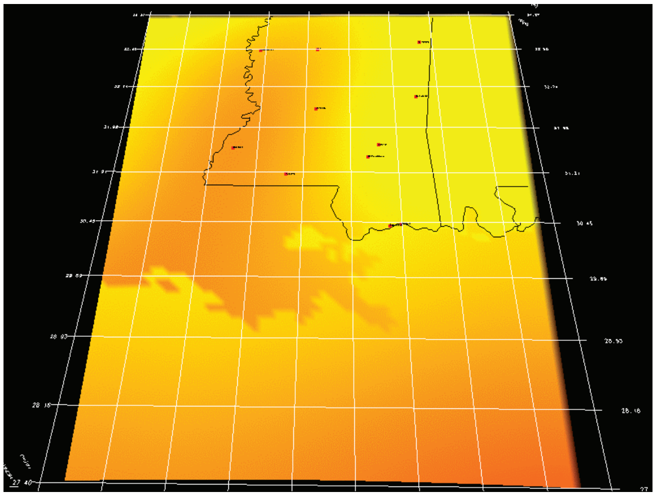

Fig. 4.5 An overview of the data set showing the geographic region.

The data sets for each configuration of the COVE are twenty-four hour forecast models from the following dates respectively: 2002 February 7, 13, 19, and 20. During all of these times, a storm of some sort was in the region.

The following variables and visualization tools were used:

- Terrain color-mapped with temperature from 250 to 300 Kelvin.
- Wind illustrated with vector glyphs, magnitude length, and magnitude color, with a color map range of 0 to 45 knots.

- •• Snow, ice, and rain with particle density fields mapped from 0 to 4 particles with a range of 0 to 0.8 g/kg.
- •• Isosurface of cloud water density of 0.4 g/kg.

4.3.4. Experiment Procedures

Before an individual user study began, the user was selected to be in the control group or the standard user group. About one-third of the users were selected to be in the control group.

First, the users were shown how to operate all of the virtual environment hardware, including the glasses, tracking, and wand. The users were then trained on the practice data set, with the training questions. The experimenter assisted the user throughout training. Since the experiments were expected to be long, there was not enough time to allow the user to practice again by himself or herself. The initial training was the only practice.

The users were given the sets of questions, with each question on a individual slip of paper. The user would take a question from the stack, in order, and attempt to answer the question using the virtual environment. The experimenter would watch the user and note when the user started and completed each question and recorded the time. Thus, it was possible to have two metrics for assessing the results of the experiment: the time it took to complete a question and the accuracy of each question. The user was not told about the timing until afterwards, to avoid stress and biasing of the user during the user studies.

After a user completed a set of questions, he or she was given the next data set in the proper COVE configuration. As discussed in Section III, it is necessary to have a control group. The control group users performed their experiments in the standard configuration of the COVE (stereoscopy, wide field-of-view) throughout the experiment.

4.3.5. Results Analysis

After all of the user studies were completed the times and accuracies had to be calculated. The times were simply averaged per configuration per question, thus yielding a table of average times for the standard users and average times for the control users. The control user averages were subtracted from the standard user averages, and an offset value was added to shift all the times into the positive range. See the formula below:

$$T_{adjusted}(q,c) = T_{standard}(q,c) - T_{control}(q,c) + T_{offset}$$

$T_{adjusted}$ is the control adjusted time; $T_{standard}$ is the time for the standard users, $T_{control}$ is the time for the control group users; T_{offset} is the constant used to shift the values into the positive range; q is the question number • [q1,…,q7]; c is the configuration • • [A,B,C,D]. T_{offset} is needed if the control group times are greater than the standard group times, which may happen due to users' individual skills. Since each $T_{adjusted}$ is to be compared only between COVE configurations, the shift is justifiable and useful. A total metric for each configuration is also computed for $T_{adjusted}$ since individual questions may not yield significant results. This metric is simply the sum of the $T_{adjusted}$ values of the seven questions:

$$T_{adjusted}(c) = \mathop{\bullet}\limits_{q=1}^{7} T_{adjusted}(q,c)$$

The accuracies were handled similarly; however, some questions had to be treated specially. In the case where questions had multiple parts (e.g., velocity and direction of wind), the accuracy of the question was the average of each part's accuracy. To calculate the accuracy, the following formula was used, derived from a relative error formula [2]:

$$A = 1 \bullet \frac{|Vuser \bullet Vcorrect|}{Vmax \bullet Vmin}$$

V_{user} is the user's answer; $V_{correct}$ is the correct answer; V_{max} is the maximum possible value; V_{min} is the minimum possible value. In summary, this calculates the difference between the user's answer and the correct answer, then scales that by the total range of possible values to a percent error and inverts that to a percent accuracy. This is similar to a relative error formula [2], but since there is a limit to the range of possible values, the range is used as the divisor instead of the correct answer.

To calculate accuracy for question seven (sky condition forecast), it was treated as an eight-part question. Each possible condition (clear, overcast, rain, hail, etc.) was a part with a value of 0 or 1: 0 for not present, 1 for present. As described above, these accuracies are calculated with the formula, yielding a 0% or 100% accuracy, and each accuracy value is averaged to calculate the final accuracy for that question.

As with the time values, the accuracies were adjusted by the control group accuracies. The following is the accuracy adjustment formula:

$$A_{adjusted}(q,c) = A_{standard}(q,c) - A_{control}(q,c) + A_{offset}$$

The $A_{adjusted}$ values are averaged for the seven questions to yield a mean accuracy for each configuration:

$$A_{adjusted}(c) = \frac{1}{7}\sum_{q=1}^{7} A_{adjusted}(q,c)$$

Finally, the resulting times and accuracies must be tested for statistical significance. The best significance test for this study is the *t-test*, since we do not know the true deviation or variance of the data [17]. To perform the t-test, we must calculate the approximate deviation for each question for each configuration. Since we want to find the significance of the adjusted time and accuracy values, we calculate the deviations with the standard and control groups combined as one group. Next, we find the t-value for the given number of subjects and a confidence percentage (e.g., 20 subjects total, 90% confidence). The t-value may be obtained from a table or a computer program. Next, significance values (per question per configuration) are computed by taking into account the t-value, the deviations, and the total number of subjects as follows:

$$S(q,c) = t\frac{\sigma(q,c)}{\sqrt{n_{standard} + n_{control}}}$$

S is the significance for each question for each configuration; σ is the calculated approximated deviation of all of the users times or accuracies for each question for each configuration; t is the t-value; $n_{standard}$ is the number of standard users; $n_{control}$ is the number of control group users. Same as the times and accuracies, the *S(q,c)* values can be summed or averaged into an *S(c)* value.

With a significance for any given configuration or question, a difference between two configurations may be labeled as significant given the confidence percentage. For example, if the time significance value for 90% confidence for configuration A is 200, and configuration B takes a total of 240 seconds less than A, then we can say the following, "performance in configuration B is significantly faster than performance in configuration A," with 90% confidence.

CHAPTER V

RESULTS

There were a total of nine users. Six of these users were standard users with the other three as control group users. The following results are only the adjusted values. For the unadjusted standard group and control group values, see Appendix C.

5.1. Performance Times

The adjusted performance times are somewhat unreliable. This is due to several factors. There are only nine subjects and random distractions and fatigue are stronger in some users than others. With more users, these discrepancies would average themselves toward insignificance. In Figure 5.1, the times of each question for configurations A-D are shown. Recall the stereo and field-of-view configurations in Table 3.1, repeated here as Table 5.1.

A	No stereoscopy, narrow field-of-view
B	Stereoscopy, narrow field-of-view
C	No stereoscopy, wide field-of-view
D	Stereoscopy, wide field-of-view

Table 5.1 (same as Table 3.1) The four configurations of the COVE.

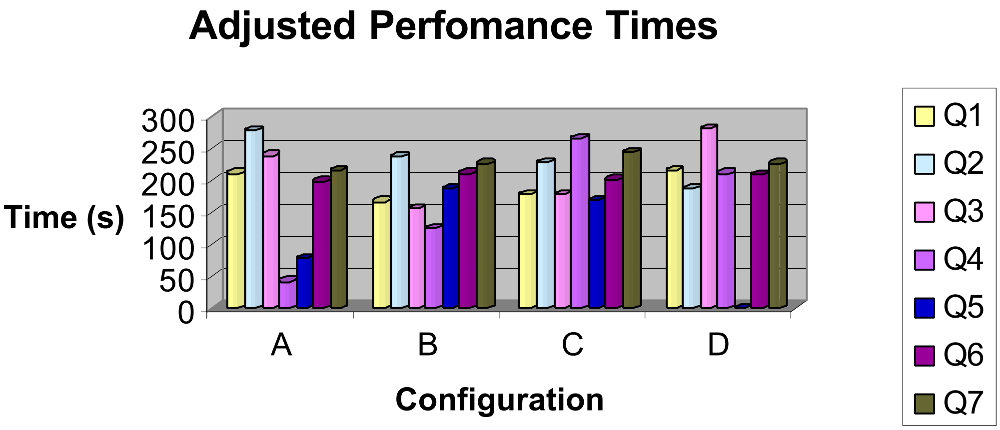

Fig. 5.1 Adjusted performance times of each question for each configuration.

Question 2 was the only question that exhibits the expected behavior. Most of the other questions decrease significantly in configuration B, then increase slightly in C, and then decrease again slightly in D. However, in Figure 5.2, the total adjusted times seem to make somewhat more sense. There is a 13% drop from configuration A to B. This is followed by and 8% increase from configuration B to C, and a 1% decrease from C to D.

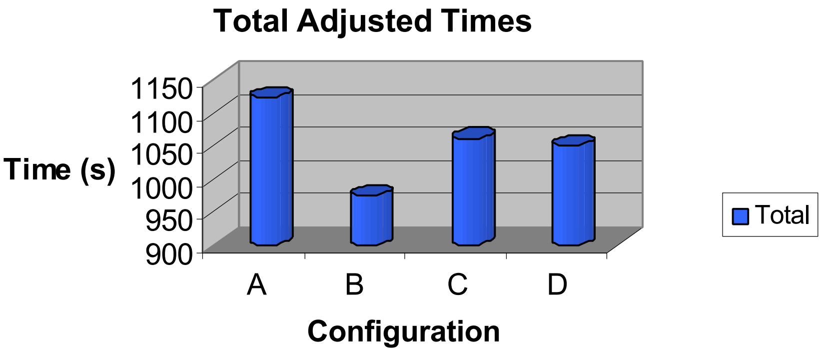

Fig. 5.2 The sum of all adjusted performance times for each configuration.

Given a 70% confidence interval, it can be shown that performance in configuration B is significantly faster than configuration A. On the other hand, this confidence renders the differences from any value to C or D statistically insignificant.

5.2 Performance Accuracies

Unlike the performance times, the accuracies are more insightful. Random distractions do not seem to radically interfere with accuracy, since the user can simply start over if interrupted. The control group seems to have factored out problems due to fatigue or question difficulty. Most individual question performance accuracies seem random, as shown in Figure 5.3.

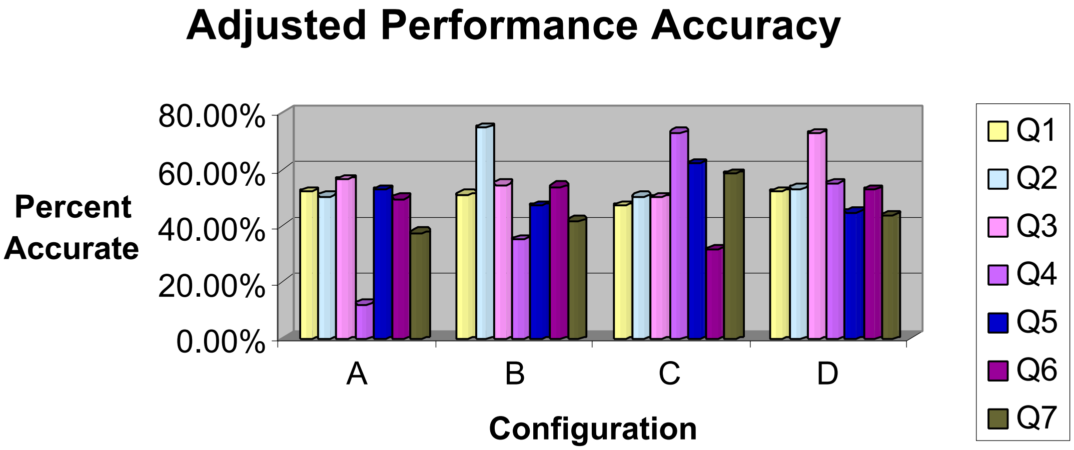

Fig. 5.3 Adjusted performance accuracies of each question for each configuration.

However, the total accuracies (actually they are averages) for each configuration indicate a significant improvement in performance due to COVE configurations. There is a 13% increase in accuracy from configuration A to B. This is followed by a 4% increase from B to C and a 0.7% increase from C to D.

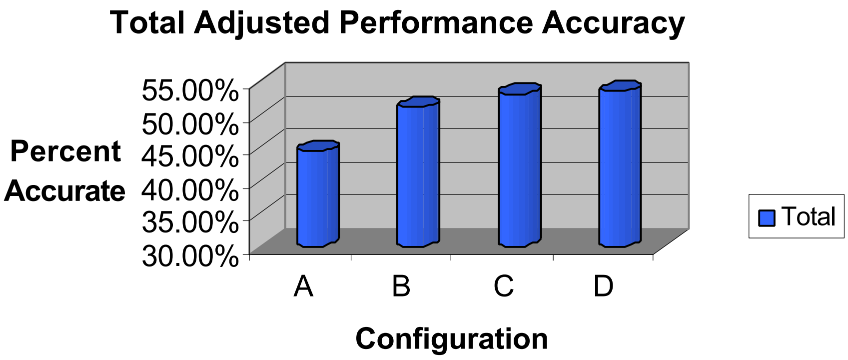

Fig. 5.4 The sum of all adjusted performance accuracies for each configuration.

With a confidence interval of 90%, the improvements from A to B, C, and D are all statistically significant. However, the differences between B, C, and D, without regard for A, are all insignificant.

CHAPTER VI

CONCLUSIONS

6.1. Discussion

In the results from Chapter V, the performance times are mostly unreliable. The only conclusion that can be drawn from them is that any amount of immersion is better than none – hence the large decrease in time from configuration A to configuration B. The smaller increases in time for configurations C and D are most likely due to interruptions and fatigue of the users. Since these factors can be random, it is difficult for the control group to balance these factors without a larger number of subjects.

The performance accuracies support the thesis that immersion, specifically stereoscopy and field-of-view, assist meteorologists in analyzing atmospheric data. The accuracy increases when using stereoscopy only (configuration B) and field-of-view only (configuration C) indicate the merits of these two immersive components. In addition, field-of-view may provide the superior improvement, but this is not supported by a confidence interval of 90%. Also, since the combination of field-of-view and stereoscopy (configuration D) isn’t significantly higher than only field-of-view, the deduction is that field-of-view provides a significant enough improvement, that stereoscopy has little effect when used in combination. However, this result may change if more users were considered.

6.2. Possibilities

Since the experiment will continue beyond this publication, more users should periodically be added to the results. Hopefully in the future, the performances will approach the expected improvements with greater confidence. For future work, a more carefully constructed experiment could improve upon this one by attempting using human factors research to approximate the factors of learning, fatigue, and interruption as random variables. This may yield stronger results, especially if the true deviation or variance due to the interfering factors could be calculated.

In addition, improving the user interface for MetVR can reduce fatigue and frustration of the users. Since the user studies were the focus of this research, and not the implementation of the software, MetVR is somewhat lacking in user interface qualities. A more powerful interface would allow users to more easily make analyses and reduce strain.

REFERENCES

[1] H. Baier, M. Buss, F. Freyberger, and G. Schmidt, "Benefits of Combined Active Stereo Vision and Haptic Telepresence," in IEEE/RSJ Proc. of Int. Conf. on Intelligent Robots and Systems, 2000, pp. 702-707.

[2] D. Berkey and P. Blanchard, *Calculus*, Orlando, FL: Saunders College Publishing, 1992, pp. 161-162.

[3] A. Bierbaum, C. Just, P. Hartling, K. Meinert, A. Baker, and C. Cruz-Neira, "VR Juggler: A Virtual Platform for Virtual Reality Application Development," in Proc. of IEEE VR 2001, Mar. 2001, Yokohama, Japan, pp. 89-96.

[4] S. Bryson, "Virtual reality in scientific visualization," in Comm. of the ACM, May 1996, Vol. 39, No. 5, pp. 62-72.

[5] K.E. Bystrom, W. Barfield, and C. Hendrix, "A Conceptual Model of the Sense of Presence in Virtual Environments," in Presence: Teleoperators and Virtual Environments, Apr. 1999, Vol. 8, No. 2, pp. 241-244.

[6] D.G. Caldwell, A. Wardle, and M. Goodwin, "Telepresence Feedback and Input Systems for a Twin Armed Mobile Robot," in IEEE Robotics and Automation, Sep. 1996, pp. 29-38.

[7] *Cave5D Home Page*, Center for Coastal Physical Oceanography, Old Dominion University, Mar. 2002, http://www.ccpo.odu.edu/~cave5d/homepage.html.

[8] *Cave5D 2.0 Home Page*, Mathematics and Computer Science & Environmental Research Divisions, Argonne National Laboratory, Mar. 2002, http://www-unix.mcs.anl.gov/~mickelso/CAVE2.0.html.

[9] B.M. Collins, “Data Visualization – Has It All Been Seen Before?” in Animation and Scientific Visualisation, British Computer Society, State of the Art Report, 1992.

[10] C. Cruz-Neira, “Applied Virtual Reality,” Course 14 Notes, at ACM SIGGRAPH 1998, Jul. 1998, Orlando FL.

[11] C. Cruz-Neira, D.J. Sandin, T.A. DeFanti, R.V. Kenyon, and J.C. Hart, "The CAVE: Audio Visual Experience Automatic Virtual Environment." in Comm. ACM, Jun. 1992, Vol. 35, No. 6, pp. 64-72.

[12] R.P. Darken, T. Allard, and L.B. Achille, “Spatial Orientation and Wayfinding in Large-Scale Virtual Spaces,” in Presence: Teleoperators and Virtual Environments, 1998, Vol. 7, No. 2, pp. 101-107.

[13] R.A. Earnshaw and N. Wiseman, *An Introductory Guide to Scientific Visualization*, Berlin, Germany: Springer-Verlag, 1992, pp. VI-VII.

[14] C. Hendrix and W. Barfield, “Presence in Virtual Environments as a Function of Visual and Auditory Cues,” in Proc. of Virtual Reality Annual International Symposium, 1995, pp. 74-82.

[15] W. Hibbard and D. Santek, "The Vis-5D System for Easy Interactive Visualization," in Proc. of IEEE Visualization, Oct. 1990, San Francisco, CA, pp. 28-35.

[16] A. Kaufman, *Volume Visualization*, IEEE Computer Press, 1991.

[17] X.R. Li, *Probability, Random Signals, and Statistics*, Boca Raton, FL: CRC Press, 1999, pp. 236-239.

[18] W.E. Lorensen and H.E. Cline, "Marching cubes: A high-resolution surface construction algorithm," in Computer Graphics, Jul. 1987, Vol. 21, No. 4, pp. 163-169.

[19] B.H. McCormick, T.A. DeFanti, and M.D. Brown, "Visualization in Scientific Computing," in Computer Graphics, Vol. 21, No. 6, (special issue).

[20] T. Möller and E. Haines, *Real-Time Rendering*, Natick, MA: A.K. Peters, Ltd., 1999, p. 1.

[21] D. Pape, "A Hardware-Independent Virtual Reality Development System," in IEEE Computer Graphics and Applications, Jul. 1996, vol. 16, no. 4, pp. 44-47.

[22] R.L. Pepper, D.C. Smith, and R.E. Cole, "Stereo TV Improves Operator Performance under Degraded Visibility Conditions," in Optical Engineering, 1981, Vol. 20, No. 4.

[23] S. Sarathy, K. Shujace, and K. Cannon, "Visualization of large complex datasets using virtual reality," in IEEE Information Technology: Coding and Computing, Mar. 2000, pp. 522-526.

[24] M. Slater and S. Wilbur, "A Framework for Immersive Virtual Environments (FIVE): Speculations on the Role of Presence in Virtual Environments," in Presence: Teleoperators & Virtual Environments, Dec. 1997, Vol. 6, No. 6, pp. 603-616.

[25] L. Smarr and Catlett, C.E. "Metacomputing," in Comm. ACM, Jun. 1992, Vol. 35, No. 6, pp. 44-52.

[26] D. Song and M.L. Norman, "Exploring Multiscale Data with Virtual Reality," in Proc. IEEE Symposium on Research Frontiers in Virtual Reality, 1993, pp. 75-79.

[27] W. Stallings, *Operating Systems: Internals and Design Principles*, Upper Saddle River, NJ: Prentice-Hall, 1998, pp. 319-360.

[28] I.E. Sutherland, “Head Mounted Three Dimensional Display,” in Proc. of the Fall Joint Computer Conference, 1968, Vol. 33, pp. 757-764.

[29] *VGeo™ Home Page*, VRCO, Mar. 2002, http://www.vrco.com/products/vgeo/vgeo.html.

[30] C. Ware and G. Franck, "Evaluating stereo and motion cues for visualizing information nets in three dimensions," in ACM Transaction on Graphics, Apr. 1996, Vol. 15, No. 2, pp. 121-140.

[31] C. Ware and S. Osborne, "Exploration and Virtual Camera Control in Virtual Three Dimensional Environments," in Computer Graphics, Mar. 1990, Vol. 24, No. 2, pp. 175-183.

[32] A. Watt and M. Watt, *Advanced Animation and Rendering Techniques*, New York, NY: ACM Press, 1992, pp. 297-298.

[33] B. Welch, “How Can We Determine if the Sense of Presence Affects Task Performance?" in Presence: Teleoperators and Virtual Environments, Oct. 1999, Vol. 8, No. 5, pp. 574-577.

[34] M.J. Wells and M. Venturino, "Performance and head movements using a helmet-mounted display with different sized fields-of-view," in Optical Engineering, Aug. 1990, Vol. 29, No. 8, pp. 870-877.

[35] G.H. Wheless, C.M. Lascara, A. Valle-Levinson, D.P. Brutzman, W. Sherman, W.L. Hibbard and B.E. Paul, "Virtual Chesapeake Bay: interacting with a coupled physical/biological model," in IEEE Computer Graphics and Applications, Jul. 1996, Vol. 16, No. 4, pp. 52-57.

[36] *WorldTookKit™ Home Page*, Sense8™ Corporation, Mar. 2002, http://www.sense8.com/products/wtk.html.

[37] C. Youngblut, R. Johnson, S. Nash, R. Wienclaw and C. Will, *Review of Virtual Environment Interface Technology.* IDA Paper P-3186. Institute for Defense Analyses, 1801 N. Beauregard Street, Alexandria, VA 22311-1772, Mar. 1996.

[38] S. Ziegeler, R.J. Moorhead, P.J. Croft, and D. Lu, “The MetVR Case Study: Meteorological Visualization in an Immersive Virtual Environment,” in Proc. IEEE Visualization 2001, San Diego, CA, Nov. 2001, pp. 489-492, 596.

APPENDIX A

METVR SOFTWARE DESIGN AND IMPLEMENTATION DETAILS

A.1. Design Chart

The chart below (Figure A.1) is an illustration of the interaction between MetVR classes and subclasses. Not all classes are listed here, only the ones that have significant impact on the overall design. Note that the CAVE-like device is only an example of a display device.

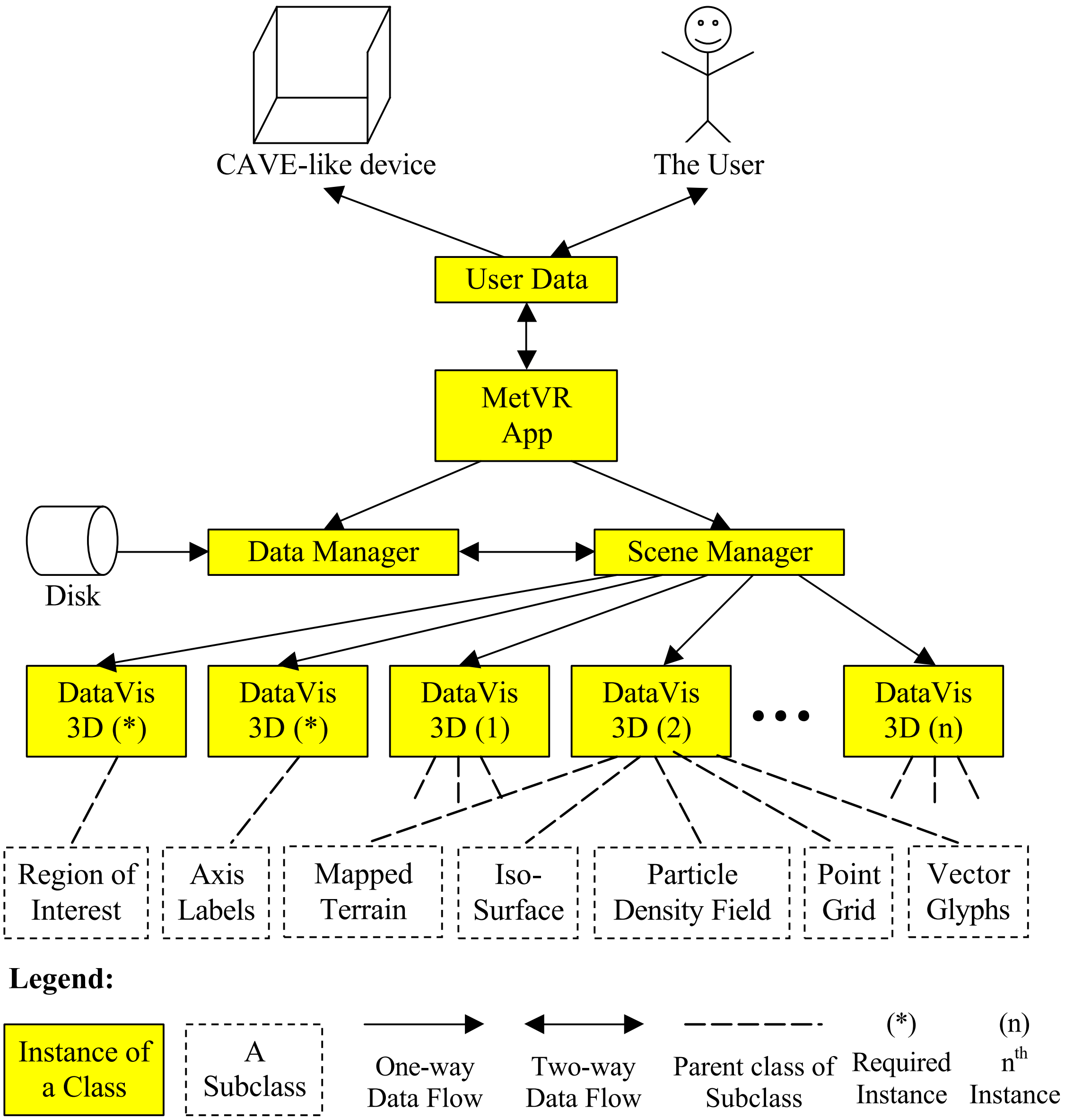

Fig. A.1 Design Chart

A.2. Design Details

MetVR is designed to be easily extensible, especially for adding new visualization tools. All of the code is written in C++ using an object-oriented design. This object-oriented design consists of a set of classes and data flow links between the classes.

As shown in Figure A.1, some of the data flow links are unidirectional (one-way) while others are bi-directional (two-way). These data flow links are not necessarily some physical connection between two classes. Some links may simply be pointers to a region of memory. So a class may actually be accessing an array somewhere else in memory, but the pointer to that memory location was received from another class. The data flow links merely indicate some sort of information was or will be passed from one class to another.

It is also worth explaining the subclass figures in Figure A.1. It should be obvious that a dotted line from a solid box to a dotted box means that the solid box is the parent class, and the dotted box is the child subclass. However, what is more complicated is that there are multiple lines from some class instances. This means that this instance of that class may be any one of the possible subclasses. The reason for this will be apparent when the DataVis3D class is explained.

The following are brief explanations of the classes in the diagram. There are five classes that will be discussed:

1. User Data
2. MetVR App
3. Data Manager
4. Scene Manager
5. DataVis3D

The User Data class handles all interaction and I/O between MetVR and the user and the display device (e.g., a CAVE-like device or CRT screen, etc.) There is one instance of this class for every user. The way the User Data instances handle user I/O and the display device depends on the VR Juggler configuration. For example, in a CAVE-like setup where there are multiple walls surrounding several users, it makes most sense for the CAVE-like displays to be controlled by only one user's head movements. Thus the primary User Data instance will control the CAVE-like device and all other users may only provide ancillary input to the program, or even be ignored. Since VR Juggler currently does nothing with secondary users, MetVR ignores input from any users but the primary user.

The MetVR App class is actually a subclass of VR Juggler's vjGlApp class. The vjGlApp class provides methods for doing pre-frame operations, drawing to an OpenGL context, post-frame computations, and other necessary functions. These functions are overridden by MetVR App to provide all of the functionality for MetVR. Most MetVR App methods consist of managing User Data instances and telling the Data Manager and Scene Manager when to update their states. Due to the nature of this class, there is only one instance.

The Data Manager handles the input data for MetVR. It handles all file format issues and disk reading operations so that the rest of MetVR is abstracted away from such functions. The Data Manager only loads the current time step into memory and is told by MetVR App when to update based on time step changes. The Data Manager also only loads fields that are currently being visualized. The Scene Manager tells the Data

Manager which fields are being used. This data is passed to the Scene Manager as pointers to arrays. The Data Manager can be rewritten or subclassed to read other formats than MetArray or even read from network streams, shared memory, etc. without affecting the rest of MetVR.

The Scene Manager class maintains all visualization scene information. The user specifies what fields are to be visualized using various tools. The Scene Manager keeps a list of DataVis3D subclass instances (the visualization tools) and which variables are associated with these instances. The Scene Manager tells all of the DataVis3D subclass instances when to perform preprocessing operations, when to draw, and when to perform post-drawing calculations. Currently two DataVis3D subclasses are required to be instantiated: Region of Interest and Axis Labels. All other DataVis3D subclass instances are managed by a list and may be added or deleted dynamically. The Scene Manager can also load and save scenes to and from the disk.

This is an abstract base class that is never actually instantiated in MetVR. It has methods for pre-draw operations, drawing, post-draw calculations, etc. Subclasses of DataVis3D override these methods to do the actual visualization. Some DataVis3D methods have default functionality if not overridden. For example, the updateGL() method builds a display list for the current time step, and the DrawFast() method draws using that display list. Most DataVis3D subclasses will utilize that functionality, thus it is not necessary to repeat the code in all of these subclasses.

The beauty of making DataVis3D an abstract base class is that the Scene Manager does not know or care which subclasses of DataVis3D it is managing. It simply traverses its list of DataVis3D instances and calls the DataVis3D methods that it knows about. Polymorphism does the work of determining which of the actual subclasses get called. This design is also advantageous in that any functionality (for example, parallel processing) added to DataVis3D is inherited by all of its subclasses.

APPENDIX B

EXPERIMENT TASK SETS

The following tables include the sets of tasks for training and the four configurations of the COVE.

1	What is the surface temperature of Greenville at 15:00?
2	What is the direction and velocity of wind 10000 meters above Laurel at 12:00?
3	What is the rain concentration value 1500 meters above 28N 88.1W at 24:00?
4	Is the total cloud concentration above Biloxi higher or lower than the total concentration above 28.5N 90W at 15:00, and by how much?
5	Where and when is the lowest temperature on the surface in this data set? Give the time. Give the longitude and latitude. Give the nearest city.
6	Can you give me a surface temperature forecast for Hattiesburg three hours after the last time step?
7	What is the sky condition forecast for Gulfport three hours after the last time step (i.e. clear, partly cloudy, mostly cloudy, overcast, rain, thunderstorms, hail, and/or snow)? List all that apply.

Table B.1 Training Tasks

1	What is the surface temperature of McComb at 12:00?
2	What is the direction and velocity of wind 1000 meters above Columbus at 3:00?
3	What is the ice concentration value 8000 meters above 29.5N 86.9W at 6:00?
4	Is the total cloud concentration above Columbus higher or lower than the total concentration above 28N 87.4W at 12:00, and by how much?
5	Where and when is the lowest temperature on the surface in this data set? Give the time. Give the longitude and latitude. Give the nearest city.
6	Can you give me a surface temperature forecast for Greenville three hours after the last time step?
7	What is the sky condition forecast for Natchez three hours after the last time step (i.e. clear, partly cloudy, mostly cloudy, overcast, rain, thunderstorms, hail, and/or snow)? List all that apply.

Table B.2 Task Set A (no stereoscopy, narrow field-of-view)

1	What is the surface temperature of Greenwood at 12:00?
2	What is the direction and velocity of wind 10000 meters above McComb at 0:00?
3	What is the rain concentration value 2000 meters above 28N 87.9W at 15:00?
4	Is the total cloud concentration above Biloxi higher or lower than the total concentration above 28.1N 87.5W at 15:00, and by how much?
5	Where and when is the lowest temperature on the surface in this data set? Give the time. Give the longitude and latitude. Give the nearest city.
6	Can you give me a surface temperature forecast for Natchez three hours after the last time step?
7	What is the sky condition forecast for Biloxi three hours after the last time step (i.e. clear, partly cloudy, mostly cloudy, overcast, rain, thunderstorms, hail, and/or snow)? List all that apply.

Table B.3 Task Set B (stereoscopy, narrow field-of-view)

1	What is the surface temperature of Gulfport at 18:00?
2	What is the direction and velocity of wind 1000 meters above Jackson at 15:00?
3	What is the snow concentration value 6000 meters above 33.5N 88.8W at 24:00?
4	Is the total cloud concentration above Greenville higher or lower than the total concentration above 31.2N 91.8W at 21:00, and by how much?
5	Where and when is the lowest temperature on the surface in this data set? Give the time. Give the longitude and latitude. Give the nearest city.
6	Can you give me a surface temperature forecast for McComb three hours after the last time step?
7	What is the sky condition forecast for Columbus three hours after the last time step (i.e. clear, partly cloudy, mostly cloudy, overcast, rain, thunderstorms, hail, and/or snow)? List all that apply.

Table B.4 Task Set C (no stereoscopy, wide field-of-view)

1	What is the surface temperature of Laurel at 12:00?
2	What is the direction and velocity of wind 500 meters above Greenville at 15:00?
3	What is the snow concentration value 4000 meters above 31.7N 89.9W at 6:00?
4	Is the total cloud concentration above Greenville higher or lower than the total concentration above 30N 92.4W at 3:00, and by how much?
5	Where and when is the lowest temperature on the surface in this data set? Give the time. Give the longitude and latitude. Give the nearest city.
6	Can you give me a surface temperature forecast for Jackson three hours after the last time step?
7	What is the sky condition forecast for Meridian three hours after the last time step (i.e. clear, partly cloudy, mostly cloudy, overcast, rain, thunderstorms, hail, and/or snow)? List all that apply.

Table B.5 Task Set D (stereoscopy, wide field-of-view)

APPENDIX C

PRE-ADJUSTED DATA

The following tables include the data recorded from the user studies before making any adjustments based on the performance times or accuracies of the control group.

AVERAGES	A	B	C	D
Q1	77	46	63	64
Q2	173	150	99	80
Q3	159	128	143	184
Q4	165	153	209	216
Q5	179	130	158	159
Q6	73	89	79	82
Q7	98	80	110	66
Total	924	776	861	851

Table C.1 Performance Time Averages

AVERAGES	A	B	C	D
Q1	66	79	85	49
Q2	96	112	72	93
Q3	120	172	166	103
Q4	322	229	144	205
Q5	301	142	190	358
Q6	75	78	78	74
Q7	82	54	67	40
Total	1062	866	802	922

Table C.2 Control Group Performance Time Averages

AVERAGES	**A**	**B**	**C**	**D**
Q1	92.11%	91.88%	90.83%	96.09%
Q2	91.39%	88.44%	88.44%	90.09%
Q3	86.34%	86.80%	55.85%	55.78%
Q4	60.29%	63.19%	65.33%	59.98%
Q5	89.69%	92.80%	89.74%	73.18%
Q6	85.00%	91.00%	66.00%	92.00%
Q7	75.00%	91.67%	91.67%	93.75%
Average	82.83%	86.54%	78.26%	80.12%

Table C.3 Performance Accuracy Averages

AVERAGES	**A**	**B**	**C**	**D**
Q1	90.11%	90.98%	93.75%	94.09%
Q2	91.02%	63.84%	88.24%	86.91%
Q3	80.14%	82.18%	55.98%	33.12%
Q4	98.18%	78.32%	42.44%	54.93%
Q5	87.05%	95.92%	78.01%	78.57%
Q6	85.33%	87.33%	84.67%	89.33%
Q7	87.50%	100.00%	83.33%	100.00%
Average	88.48%	85.51%	75.20%	76.71%

Table C.4 Control Group Performance Accuracy Averages