Visual Analytics of Multidimensional Data Using Immersive High Performance Computing Center Data

by

Lino Abiel Virgen Gracia, A.A.

A Thesis

In

Computer Science

Submitted to the Graduate Faculty of Texas Tech University in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCES

Approved

Dr. Tommy Dang
Chair of Committee

Dr. Susan Mengel

Dr. Mark Sheridan
Dean of the Graduate School

May, 2019
Acknowledgements

First of all, I would like to thank Dr. Tommy Dang for introducing me to virtual reality and data visualization. I would like to thank him for guiding me through this thesis project, providing vast expertise knowledge, finding opportunities for me to improve my work, and always pushing me to do my best. I would also like to thank the people at the iDVLab at TTU for providing endless feedback and support.

Additionally, I would like to express my infinite gratitude to the Terry Foundation not only for fully funding my undergraduate and graduate degrees but for allowing me to be part of a bigger family that focuses on mutual advancement and support.

Also, I would like to extend my gratitude to Mrs. Uma Puri for introducing me into the world of computer science. Although she taught me programming, she also taught me discipline, resilience, and principally, the desire to always aim for success, for which I will be forever grateful.

Finally, I would like to thank my parents for always believing in me, teaching me to never quit, inspiring me to pursue higher education, and providing needed and comforting counsel.
Table of Contents

Acknowledgements ........................................ ii
Abstract ................................................. v
List of Figures ........................................ vi
1. Introduction ........................................... 1
   1.1 Purpose ........................................ 1
   1.2 Outline ....................................... 1
2. Background ........................................... 2
   2.1 Multidimensional Data ............................ 2
   2.2 Data Visualization vs Data Analytics ........... 3
   2.3 Virtual Reality ................................ 5
3. Implementation ....................................... 8
   3.1 Introduction ................................... 8
   3.2 A-Frame ....................................... 8
   3.3 Three.js ..................................... 9
   3.4 Dataset ...................................... 10
4. *HiperVR*: An Immersive Visualization for HPCC ....... 11
   4.1 Physical Representation ......................... 11
   4.2 Control Panel .................................. 12
   4.3 Host Dossier .................................. 15
   4.4 3D Scatter Plot Matrix ......................... 16
   4.5 Logic Operations ............................. 20
   4.6 Controls ..................................... 20
   4.7 Challenges ................................... 21
Abstract

In the era of big data, visualizing data techniques have had to modernize in order to cover the increase in data volume and dimension. The recent explosion in popularity of virtual reality technology has provided an immersive platform for users with a vast number of opportunities for data not only to be visualized but also to be interacted with. This project introduces HiperVR, a virtual reality environment that simulates the High Performance Computing Center at Texas Tech University with a new data analysis visualization tool. This system simulates the server room on campus and provides data insight on each node. The data analysis visualization tool is a 3D scatter plot matrix able to represent relationships of up to 4 attributes. This visualization implements Scagnostics, a data analysis tool, in order to find data features in each scatter plot. Using data features, such as outlying and monotonic scores, users are able to filter the matrix and focus on the scatter plots that they are interested in. Thus, HiperVR serves as an application of the integration of data analysis visualizations into a virtual reality simulation.
List of Figures

2.1 *ComplexHeatmap* ........................................... 3
2.2 *GPLOM* .................................................. 4
2.3 Visual analytics examples ................................. 6
2.4 Immersive visual analytics examples ................. 7
3.1 *HiperVR* .................................................. 9
4.1 Quanah nodes in *HiperVR* ............................... 11
4.2 *HiperVR’s* control panels ............................... 13
4.3 Racks changing colors ................................... 14
4.4 Host dossier view ......................................... 15
4.5 3D scatter plot matrix ................................... 16
4.6 Scatter plot matrix transformation ..................... 17
4.7 Filtered scatter plot matrix .............................. 18
4.8 Individual scatter plot ................................... 19
5.1 *VRParallelSet* ............................................ 24
A.1 Scagnostics measures ................................. 32
Chapter 1

Introduction

1.1 Purpose

The purpose of this project is to introduce a data analysis visualization that is able to provide an in-depth understanding of multidimensional data. Using the High Performance Computing Center (HPCC) at Texas Tech University as a case study, HiperVR simulates the physical building and provides data visualizations on each of their servers. The system also provides data analysis on the health of each host utilizing a 3D scatter plot matrix and data analysis tools like Scagnostics.

1.2 Outline

This thesis is divided into several chapters. Chapter 2 introduces major concepts on data science and lays out some of the recent advances of data visualization analysis and its virtual reality applications. Chapter 3 describes the hardware and software used to implement the presented system. Chapter 4 explains in detail the features of HiperVR. Chapter 5 reports some potential improvements and features for HiperVR as well as additional visualization ideas. Chapter 6 provides the conclusion for this thesis. Finally, the Appendix includes additional definitions and concepts.
Chapter 2

Background

This chapter will introduce multidimensional data and some recent studies on it. It will also describe data visualization and data analysis. Finally, it will report recent works on virtual reality and how it has helped in the data analysis field.

2.1 Multidimensional Data

Multivariable data, or multidimensional data, is a set of data points with 3 or more attributes (1), and it is used everywhere. From social science (2) (3) to medicine (4) (5) to natural environment analysis (6) (7) multidimensional data is almost unavoidable to use. For developers utilizing this type of data requires some forms of manageable formats. One of the most famous forms of storing any sort of data is using a relational database, conceived by E.F. Cold back in 1970 (8). However, due to the continuous increase in popularity of web applications, simpler and lighter formats are needed. JavaScript Object Notation, or JSON, was introduced by Douglas Crockford in 2001. It is a serialized format, which in contrast with logical data models, can represent complex, nested, and schema-less data (9).

Visualizing multidimensional data allow users to not only to analyze the set of data points but also the relationship between attributes. There are several techniques used nowadays to visualize multidimensional data. One of them is scatter plot, or more specifically, scatter plot matrix which contains all the pairwise combinations of variables in matrix (10). Another is parallel coordinates which uses an axis per variable and creates a path per point by connecting each axis at different heights (11). Finally, there are others like glyph plots, arc-diagrams, heatmaps, hierarchies, etc (12).

Recently, there have been new advances in multidimensional data visualization techniques where they have aimed to increase the amount of data volume,
variety, and dynamics. One instance is *ComplexHeatmap*, a package that provides heatmap customization, annotation graphics, and helper parallel heatmaps in order to find trends and correlations in multivariable data (13). An example visualization is shown in Figure 2.1. Moreover, *GPLOM* combines scatter plots, heatmaps, and bar graphs to give an overview of categorical and continuous variables (14). The developers of *GPLOM* found that, even though their application is successful at analyzing multidimensional data, it has serious drawbacks in terms of interface interactions. Utilizing only the mouse to interact with the plots, they concluded that there was 'lack of clear affordances for interaction" (14). Possibly this might be due to the fact that there are few controllers. An example of *GPLOM* is illustrated in Figure 2.2. One last example is *DICON*. This tool visualizes multivariable data clusters through the use of icon images and forms treemaps for statistical evaluation (15). The developers found, through a use case study, that users enjoy the ability to drag-and-drop. They also found that using icons is helpful when representing abstract data. However, the interviewees mentioned that they would like more analysis and comparison between data attributes (15).

![Figure 2.1. ComplexHeatmap example visualizing microbiology data (13).](image)

### 2.2 Data Visualization vs Data Analytics

It is crucial to understand the distinctions between data visualization and data analytics which, even though they overlap, are their own independent fields.
Data visualization can be anything that represents data visually and more specifically information visualization is used to "represent data sets that don’t have inherent spatial components" (16). Some examples of data visualization include basic line graphs, area graphs, bar graphs, pie charts, etc.

In the other hand, data analytics is the field that "analyzes data sets and draws conclusions using the information that they provide" (17). When data analysis is supported by data visualization the product is called visual analytics. According to Duke University, visual analytics is "the practice of using visualizations to analyze data" (16). As data becomes increasingly complex, data analysts require visual aids to interpret relevant information. Hence, visual analytics relies on computational analysis and the human’s "flexibility, creativity, and background knowledge to gain insight into complex problems" (18). Data visualizations, for instance, can be used to see trends and patterns in the data. Some visualization used for data analytics are scatter plots, networks, box plots, etc. Figure 2.3 illustrates the mentioned visualization examples.

One major data analysis visualization is the scatter plot. Each data point
is made up of two or more attributes and by plotting all of the points in a grid a picture of the overall trend is created. Hence, users can get plenty of information metrics by analyzing the position of such points. Some of these metrics are shown in Figure A.1.

2.3 Virtual Reality

Virtual reality (VR) is medium through which real or fictional objects are visualized in a virtual space giving the illusion they are physically real (22). It has wide use in the areas of gaming (23) (24), multimedia (25), simulation (26) (27) (28), and others (29). Applications of VR are also present in data visualization. In fact, there have been many recent advances in the field. One of them was achieved by Andrew Moran et al. by trying to improve big data analytics. They developed a geospatial environment of MIT’s campus to represent Twitter data (30). They employed dot plots to understand the volume of tweets in geographical areas. Another recent development is vCaltech (31). vCaltech is a data visualization tool which allows the user to navigate through a 3D scatter plot which visualizes an 8-dimensional dataset. The visualization also maps 3D objects to additional information that cannot be visualized. It can be seen in Figure 2.4 a). Finally, once again due to the ease of accessibility demand, some new immersive data visualization tools have been developed for the web. An instance of such was developed by P. Butcher and P. Ritsos in 2017. They developed a virtual reality web application using Three.JS that displayed multivariable data through 3D bar charts (32) as seen in 2.4 b). In addition, they found that their visualization compromises the processing power of a browser.
Figure 2.3. Data visualizations used for data analytics. a) Box plot used to find variance and outliers (19). b) Network graph used to find relationships and associations between data points (20). c) Scatter plot used to analyze relationships and associations between attributes (21).
Figure 2.4. Immersive data analysis visualization examples. a) vCaltech allow users to navigate inside of a multidimensional scatter plot (31). b) VR web application that shows an interactive 3D bar char (32).
Chapter 3
Implementation

This chapter lays out the hardware and software tools utilized to develop HiperVR. It explains the strengths and weakness of each software library and why they were used.

3.1 Introduction

HiperVR was developed in the Interactive Data Visualization Lab, or iDVLab, at Texas Tech University utilizing an iMac (Retina 5K, Late 2015) 3Ghz Dual-Core equipped with a standard 27-inch screen of resolution of 2,560 x 1,444 pixels and an Oculus Rift with web browser Supermedium installed (33). HiperVR is a web application hosted in GitHub (34) and accessible at idatavisualization-lab.github.io/HPCC/HiperView/HPCC_VR/. Its development process began in September 2018. Its framework was based on the HyperView web application. The HyperView was also developed in the iDVLab by Ngan Nguyen and Tommy Dang and it provides a 2D visualization of the HPCC using the D3.js library extensively (35). Figure 3.1 shows the overall picture of the system’s virtual environment.

3.2 A-Frame

HiperVR is a web application which uses A-Frame as the main VR framework. A-Frame 0.8.2 is a HTML library that provides 3D and VR creation tools for web applications (36). Given the native virtual reality functionalities of A-Frame, the whole JavaScript application was built on top of an A-Frame scene. Also, since A-Frame is all written in Three.JS, it was easy to incorporate the dynamic Three.JS portion, which is the majority of the system, to the A-Frame scene. It is indispensable to point out that A-Frame utilizes HTML to generate 3D objects and scenes thus making such generations static when loaded. In other words, the generated objects can only be modified by identifying its DOM element and bypassing multiple levels of hierarchical JavaScript objects. This would
Figure 3.1. Virtual HPCC with data visualizations as represented in *HiperVR*.

have made the already computational-heavy processes heavier and slower. That being the case, only a few scene related objects were created statically, such as the actual scene, lights, and the VR controllers.

3.3 Three.js

The Three.js library was used to create all of the objects inside the A-Frame scene. Three.js is a JavaScript library that provides a framework to create 3D and 2D objects (37). The dynamic nature of JavaScript allowed these objects to be created, modified, hidden, and showed at different times throughout the applications’ lifespan while it also provided the web application’s crucial responsive behavior. Although Three.JS requires more line of codes to generate an object compared to A-Frame, it grants the developer greater customizability. Namely, the ability to use accurate measurements and coordinates on every object consistent among the whole scene was undoubtedly required for the virtual room and the various graphs.
3.4 Dataset

The High Performance Computing Center (HPCC) provides high-performance computing and visualization for the Texas Tech University population (38). One of its main clusters is *Quanah* which has 467 nodes each with 36 cores. Currently, the health of each node is monitored using a Nagios Redfish API integration (39). While Nagios is an industry tool that monitors hardware system components (40), Redfish API provides simple and secure management of hardware platforms (41). By accessing the HPCC’s Nagios server, each node’s health information (temperature, fans speed, memory load, CPU load, and power usage) can be obtained in real time. The dataset utilized in *HiperVR* is a series of snapshots taken on this health status data saved in JSON format.
Chapter 4

HiperVR: An Immersive Visualization for HPCC

HiperVR is intended to provide the user with an immersive experience of the High-Performance Computing Center at Texas Tech University while adding data analysis enrichment. With this in mind, the VR room encompasses distinct sections that deliver different aspects of the host’s data. Such sections of the VR room are the host racks, the control panels, the host dossier, and the scatter plot matrix.

4.1 Physical Representation

Figure 4.1. Quanah nodes arrange per rack at the front of the room.

The physical representation of HiperVR is made up of the room and a series of racks at the front of the room. They provide the virtual representation of the "real" or physical server room at the HPCC building. They serve two purposes. First, they allow the user to intuitively match graphical notations and objects that are purely virtual to the real world. In other words, the user is more likely to feel familiar with the environment and assume the rest of the virtual room to be an extension to the real one (42). Secondly, the physical representation of the room can be used as a reference point for navigation. This is crucial since most of the virtual room is made up of "floating" objects and navigating or interacting with them can cause motion sickness (43).

The room, which consists of a floor, a ceiling, and 4 walls, aims to give the
user a sense of familiarity and a reference point for navigating. Accordingly, all of the sides of the room are consistent with the actual server room. The floor even depicts the vents that are used to keep the room cool. Moreover, lighting in the room is also provided in order to create the shadows and shades for all the objects in the scene.

In the front of the room there are 10 racks each with a maximum of 60 hosts. Each host has exactly 2 CPUs. They represent the actual nodes of the HPCC’s Quanah cluster. As noticed in Figure 4.1 in rack 7, not all servers are available thus no host frame is drawn and the space is left empty. However, if the information is provided in the JSON file by the Redfish, each individual CPU will be colored according to the CPU or host reading in the selected service. The color gradient on the scheme utilized in HyperView. Figure 4.1 illustrates a collection of hosts colored based on the CPU’s temperature reading. It can be observed that the last host in rack 10 is not available as its colorless. At run-time, all available hosts are colorless and they begin getting their respecting color as data from the JSON file is loaded per timestamp.

4.2 Control Panel

In order to create an immersive and, particularly, a responsive environment, HiperVR provides a control panel section made up of 3 individual controllers for various features. All these controllers are put close together for the user to reach them rapidly. The time controller is the controller on top. It resembles a clock, however, the number represents each of the timestamps recorded on the JSON file. At runtime, the time controller highlights the first timestamp and moves to the next timestamp as the system finishes updating the racks and scatter matrix sections. Also, the user can click on whichever number on the controller and the system will process and visualize the data for that timestamp. It is shown in the top section in Figure 4.2.

The time controller was created using a triangular prism for each timestamp to form a circle. The prism can be click on to select that particular timestamp. When the prism is clicked it will move slightly forward and it change its
Figure 4.2. HiperVR’s control panels. Top: Time control panel. Middle: Service control panel. Bottom: Scagnostic metric control panel.

color to green. If another prism in the controller is clicked the previously highlighted prism will returned to its original position and color. In the middle of the circle there is an inactive "REAL TIME" button. It is intended change the loading data from sample to real (more information on Chapter 5. Moreover, a rotation movement along the time controller’s z has been added. The purpose of the rotation motion is mainly for the user to experiment a feeling of dynamism when using the application, and avoid a static image-like environment feeling.

Furthermore, the second controller, located in the middle, controls the selected service among the whole room visualization. This selected service, as mentioned previously, is denoted in both racks and scatter plot matrix as the color scheme; usually red for high value and blue for low value. An example of this can
be seen in 4.3 where racks 1-5 show the coloring for "Temperature" and racks 5-10 show the coloring for "Memory_load." This controller also provides interaction by clicking on the image. That will consequentially highlight the selected service and update all visualizations host by host. It is illustrated in the middle section in Figure 4.2.

The service controller was created using a 2D panel for each service provided. Even though there are 7 services (attributes) per data point, only unique attribute types are counted for this controller. The reason for this is because if the attribute types are similar they can use the same color scheme; such is the case with "Temperature" 1 and 2, and "Fans_speed" 1 and 2. Likewise, clicking on a panel will highlight it. The icons used are the same as the ones used in the scatter plot matrix labels so that the user is able to relate the representation of each data attribute.

Finally, the last controller is the scagnostic metrics filtration located at the bottom. This controller enables the user to filter the scatter plot matrix by a given set of data metrics thresholds. The metrics provided are outlying, clumpy, convex, monotonic, skewed, skinny, sparse, striated, and stringy scores as shown in Figure A.1. A slider is used to set the minimum score for each metric to filter the scatter plot matrix. The bottom section in Figure 4.2 is an instance of filtering the scatter plot matrix by the outlying score where only the scatter plot with an outlying score greater or equal to 0.4 are to be shown.

The scagnostic filtration interface was created similar to the service controller. It consists of 9 panels, one for each scagnostic metric. Each of these
panels has its own slider to set the threshold. The threshold represents the ratio of whichever metric the user wants to focus on. For instance, if the slider is in \textit{covex\_score} is on 0.4 it would mean that the user wants to view all scatter plots with a convex score equal to or greater than 0.4. The controller also has up and down arrows to rotate itself. At start, at least one slider is not set to 0 to avoid rendering all scatter plots at once.

4.3 Host Dossier

![Host dossier view](image)

\textbf{Figure 4.4.} Host dossier view. Top shows temperature changes of CPU1 and CPU2. Bottom shows health readings of current timestamp (35).

The host dossier is an individual host visualization mainly in 2D that provides an insight into a host’s health status. This visualization is an import from \textit{HyperView} and it is produced using the D3.js library. It can be viewed by clicking on a host on the racks. It will appear in a vertical panel on the right side of the racks and it will rotate following the user as he/she navigates the room. The panel is divided into two. The top section is a line graph that represents the two CPU’s temperatures in each timestamp. Below, a radar chart depicts the
metric for each service. If the user clicks on another host on the racks this panel will update with the corresponding data. Figure 4.4 shows the dossier of host 17 of rack 9.

4.4 3D Scatter Plot Matrix

![3D Scatter Plot Matrix](image)

**Figure 4.5.** 3D Scatter plot matrix visualizing 3 axis combinations of a dataset with 7 attributes.

The last and major section of *HiperVR* emphasizes the data analytics aspect of the system. Located in the back of the room, a 3D scatter plot matrix includes a series of 3D scatter plots. Each scatter plot combines 3 services of the 7 total services; they are 'Temperature_CPU1', 'Temperature_CPU2', 'CPU_Load', 'Memory_Load', 'PowerUsage', 'Fan_Speed1', and 'Fan_speed2'. Henceforth, based on the combinatorics formula

\[
\frac{n!}{(n-r)!}
\]

, where \(n\) is the number of services and \(r\) is the number of axes per scatter plot, there is a total scatter plot count of 35 each with a distinct axis combination of 3 services. Figure 4.5 is the initial scatter plot matrix with no filtration. An individual scatter plot truly visualizes up to 4 attributes since the data points represented in the plot are colored based only the selected service.

In order to provide a cohesive and macro-level view, the matrix layout was based on the 2D scatter plot matrix layout. An example of such is shown in Figure 4.6 a) where the black squares represent the scatter plots and the red squares represent the attribute label. The converted 3D layout is shown in Figure
Figure 4.6. Scatter plot matrix transformation. a) 2D scatter plot matrix. b) 3D scatter plot matrix.

4.6 b). However, the HiperVR layout is a break up of the 3D converted layout, shown in 4.5. The reason to break it up along the z axis is to prevent overlay and obstruction since they are common disadvantages of 3D graphs (44). Furthermore, the 2D matrix uses a diagonal with labels for both \( x \) and \( y \) axes rather than using horizontal and vertical lines for labels, hence HiperVR uses the same concept. Also, each \( z \) axis label is placed on the corner of every \( z \) metric subplot and is turned 90 degrees in order to maximize the intuitive readiness and comprehension of the matrix. It can also be observed in 4.5 that icons are used instead of text labels to not only prevent the user’s field of view to become crowded but also to reduce the number of objects which in return reduces the rendering time of the system.

In order to filter the scatter plot matrix the user can utilized the Scagnostic metric control panel described in Section 4.2. When a scatter plot is filtered it disappears leaving a black placeholder, as shown in Figure 4.7. A scatter plot cannot be displayed if one of the attributes required is not found in the dataset. Figure 4.5, henceforth, has a few scatter plots missing.

Similarly, another filtering capability is provided directly in the scatter plot
Figure 4.7. Scatter plot matrix filtered by outlying scores greater than or equal to 0.2.

matrix. Looking at the whole matrix, finding the scatter plot which contains a desired combination of attribute axis might be a difficult task. Hence, the user is able to click on any scatter plot matrix label to show only the scatter plots that visualized that attribute. Once clicked, all non-related plots are hidden. This includes the one filtered by the scagnostic filtering panel. In addition, other labels can be pressed to apply a more specific filtering. In order to reset this kind of filtering the user has to interact with the scagnostic filtering panel.

Another individual scatter plot’s functionality is that it allows the user to take a deeper view of the relationship of 3 distinct services. The user is able to click on a particular plot which will then move to the height view of the user. This enables the user to get closer to a plot and see the individual hosts move from timestamp A to timestamp B. A visual is provided in Figure 4.8 which shows a close up of a scatter plot where the selected service is 'Temperature'. Thus, it can be observed that the color accurately represents the axis label 'Temperature2'. In that instance, most hosts are located between 30 and 60 degrees of temperature. The user can also click anywhere else in the room to return the selected scatter plot to its original position.

The scatter plot matrix is created using a nested loop, one loop for each matrix dimension. The loop runs as many times as attributes in the data. When an attribute combination is unique, a scatter plot would be created at the 3D position represented by the 3 attributes in the combination. The resulting matrix is the same as the one shown in Figure 4.6 b) with scatter plots instead of black cubes. Then, all scatter plots in each individual z-axis are moved along the x-axis.
Figure 4.8. Individual scatter plot showing 'Power_usage', 'Temperature2', and 'CPU_load'. The data points' colors also represent temperature values.

until there is no obstruction between each other. Finally, all plots’ z coordinates are set to the same value. A similar but separate process is performed for the attribute labels.

A scatter plot is formed using the outline of a cube to denote the space of the plot. Each axis has a set of intervals. These intervals are based on the highest and lowest non-empty values of each attribute. The number of intervals per attribute can be easily increased or decreased inside the source code. The axes of the scatter plot also have labels to identify the attribute they represent. Furthermore, each data point is visualize using a sphere. The location for each sphere is obtained using a mapping of the scene space into the scatter plot space. When a sphere’s location changes, that same mapping algorithm is applied.
4.5 Logic Operations

The dataset is in JSON format and it provides the health status for all 467 hosts. Each health status is an array of length equal to the readings that were taken. At runtime all hosts in the racks are black and all data points begin at (0,0,0) in their corresponding scatter plot. Using the interval JavaScript function, one host’s information is loaded at a time. When one host’s information is loaded its virtual host gets colored and the data point of each scatter plot representing such host is moved to its proper location and is also colored. If one of the point’s visualized attributes is null the attribute will be marked as 0. As other timestamps are loaded colors and data point positions are updated. If the user clicks on a specific timestamp using the time controller the auto-update process will stop.

As soon as the system finishes loading all host information of a timestamp, Scagnostics is applied to all scatter plots in the matrix. Scagnostics is a library that provides features of sets of data with multiple attributes (45). These features include but are not limited to outlying score, monotonic score, convex score, and sparse score. Getting the outlying score of a scatter plot, for example, can help the user identify malfunctioning hosts or possible trends among certain health readings. Running 35 simultaneous scagnostics calls is a computational heavy task. Hence, JavaScript promises are used to provide parallelism and speed up to this process.

When a scagnostic filtering is applied all scatter plots’ scagnostic metrics are inspected. If one of the scagnostic metrics do not match the filtering controller the scatter plot will begin the process of hiding, otherwise it will reappear or stay in the screen. Similarly, this whole process is performed when the new timestamp information is fully loaded.

4.6 Controls

HiperVR is available in 3 platforms: Desktop, Oculus Rift, and Google Cardboard (All of them accessible from through the web). Hence, in order to interact with the system there 3 different but simple set of controls. Firstly, the desktop version utilizes the mouse and the keyboard. To click on the screen the
user can simply use the left button on the mouse. And to move the camera the user has to click on the screen and drag it to the desire angle. Also, the user is able to navigate through the scene using the AWSD keys on the keyboard.

Secondly, the user can access the system using the Oculus Rift. The user must have the application *Supermedium* installed as it serves as an internet browser capable of loading A-Frame sites (33). Once the user has accessed the website, he/she can click on the screen using the trigger button on the right hand controller. To change the camera angle, the user only need to move his/her head. Furthermore, the user can walk throughout the room to navigate the scene. The room’s objects as put close together as one of the reasons is the limited walking distance in the Oculus.

Finally, the user is able to access *HiperVR* using a mobile device with a browser installed. Using the Google cardboard’s unique button, the user can click on the screen. And to navigate the user must hold that button down. To change the view’s angle, the user can also move his/her head.

### 4.7 Challenges

Since most of the virtual environment is generated at run-time there is a small delay when the page is loaded. Similarly, hundreds of objects (control panels, racks, scatter plot matrix) are continuously updated. Such update includes recoloring, rotations, instantaneous translations, and continuous translation. The difference between the last two is that objects can be natively be moved instantaneously from one position to another using Three.JS. However, in order to create a more realistic and soothing visualization, many of the objects move gradually to their target position using an interpolation algorithm. This heavily impacts the rendering process of the browser. In fact, there is a noticeable lag in the image when many objects are moving at the same time.

Additionally, some data processing functions are directly imported from the *HiperView* as well as some functions concerning the host dossier. Thus, there has been constant updates to keep the compatibility between both systems. Having those functions come from *HiperView* is beneficial since both system are supposed
to work together to provide a wholesome view of HPCC. Nevertheless, unexpected HiperView updates do tend to cause certain features of HiperVR to malfunction, particularly the host dossier.

One early goal of this system application is to be scalable. Beginning with the number of recorded timestamp, the system is be able to adapt to the dataset size. The number of timestamps is hence represented by the time controller and less obviously by the time it takes the system to go through all timestamps in the auto-update cycle. Also, some data characteristics have changed recently. One example is the amount of dimensions/attributes per data point. The new JSON files now contain a third CPU which increases the number of temperature and fan speed readings. Although those changes are not seen in the web application, the system was develop to handle any number of attributes. Those changes should be visually noted in the control panel (service controller) and on the scatter plot matrix. However, since the scatter plot matrix highly impacts the processing power of the browser, an old dataset with fewer attributes is utilized in the web app. Similarly, adding other 3D objects to the system should also be relatively easy to do thanks to the system’s Three.js use.
Chapter 5
Future Work

This chapter provides feedback to researches who would like to improve this project. It also gives suggestions on alternative potential visualizations for this or other projects.

5.1 Improvements

In order to reduce waiting time when the page is initially loaded some of the static objects could be created by web workers or JavaScript promises. This will allow dynamic objects to be created faster and the page to load quicker regardless of the static objects that are not needed for the page to start running. Moreover, the inclusion of parallelism to any web application with a heavy processing load is unavoidable. Nevertheless, one must still consider the send and receive messages between multiple threats as they can still slow down the application.

Adding another control panel or input text box to find a desired scatter plot among the matrix is likely to help users facilitate their interactions with it. This might also be useful to reduce rendering time. Since currently the user has to look around the matrix to find a certain label and then apply the filtering/hiding process to the scene, focusing on a static screen will allow the browser to compute and process first and render visuals later.

Moreover, doing a user study is indeed a priority when it comes to the direction of this application. Having a thorough interview session with domain experts will produce highly crucial feedback and potential suggestions when it comes to data representation and data interactions. Additionally, interviewing data analysts would also be helpful. They would certainly provide a honest evaluation of the 3D scatter plot matrix and provide suggestions for other visualization techniques.
5.2 Additional Features

As far as additional features, one priority would be to bin all of the scatter plots. Binning in a scatter plot reduces the number of data points while keeping an accurate representation of the overall data, depending of course on the number of bins used. Hence, by using this method the rendering time would drastically decrease and the user would perhaps enjoy a less crowded visualization. In addition, having fewer data points would alleviate the burden of Scagnostics to compute high-volume data.

Another future feature could be the implementation of other multidimensional data visualizations. Utilizing a 3D parallel coordinates plot, for instance, might provide more useful data like seeing the proportion of hosts with low, medium, or high power usage. An example of an actual VR parallel coordinate application is shown in Figure 5.1.

![VRParallelSet: 3D parallel coordinates implemented in a VR web application (46).](image)

**Figure 5.1.** *VRParallelSet:* 3D parallel coordinates implemented in a VR web application (46).

Given *HiperVR’s* current application, it would be immensely beneficial to also add a real-time modality. In order to efficiently monitor hosts’ health information in real time the system would have to be connected to the Nagios server so it can get constant data. Having this would allow HPCC personnel to monitor servers remotely. If one of the host were to malfunction, he or she could observe it immediately in *HiperVR* and report it. Also, he or she could observe changes on other hosts or services with respect to the malfunctioned host using the
scatter plot matrix. This indeed is highly doable due to the fact that HiperView already has this modality.

Finally, a promising direction for this project is Augmented Reality (AR). In this case, the user would be able to go to the actual server room with his/her mobile phone, point the phone’s camera towards a node, and see real-time health data. Similarly, it would be able to show data analysis visualizations on top of the real physical room. An advantage of this would be a quicker mapping between the data and the real world.
Chapter 6

Conclusion

Virtual reality has had a significant impact on data visualization. It expanded not only a visual dimension but the way people interact with data. Recently, people have been able to navigate around bar charts and scatter plot while getting additional data through interaction. However, there is a limit to the information users can obtain from basic graphs, especially if the dataset to be visualized is multidimensional. Providing data analysis on top of a data visualization has helped users get a better understanding of high volume and high dimension data. Thus, HiperVR is introduced in this thesis as an application of the integration of data analysis visualizations into a virtual reality simulation. HiperVR contains a virtual room with the servers at the High Performance Computing Center at TTU. These virtual servers provide color-coded information on their health status plus a detailed time series graph. Furthermore, it contains a 3D scatter plot matrix able to visualize up to 4 attributes per data point. Using 3 axes and a color scheme, this visualization helps the user see an overall trend and relationships between the 7 attributes of the dataset. The user is able to interact with the matrix by filtering according to data features, such as outlying score, and to focus on a desired scatter plot. Altogether, HiperVR aims to implement a new multidimensional data visualizing technique which exploits VR technology and supports in-depth data analysis.
References


Appendix
Figure A.1. Examples of each scagnostic measure (47).