Improving Big Data Visual Analytics with Interactive Virtual Reality

by

Andrew Moran

B.S., Massachusetts Institute of Technology (2014)

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

Master of Engineering in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2016

Author . . . .
Department of Electrical Engineering and Computer Science
January 29, 2016

Certified by. . . . .
Vincent W. S. Chan
Professor of EECS and Aeronautics and Astronautics
Thesis Supervisor

Certified by. . . . .
Jeremy Kepner
MIT Lincoln Laboratory Fellow
Thesis Supervisor

Accepted by . . . .
Dr. Christopher Terman
Chairman, Masters of Engineering Thesis Committee

Improving Big Data Visual Analytics with Interactive Virtual Reality

by

Andrew Moran

Submitted to the Department of Electrical Engineering and Computer Science
on January 29, 2016, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Computer Science and Engineering

Abstract

For decades, the growth and volume of digital data collection has made it challenging to digest large volumes of information and extract underlying structure. Coined 'Big Data', massive amounts of information has quite often been gathered inconsistently (e.g from many sources, of various forms, at different rates, etc.). These factors impede the practices of not only processing data, but also analyzing and displaying it in an efficient manner to the user. Many efforts have been completed in the data mining and visual analytics community to create effective ways to further improve analysis and achieve the knowledge desired for better understanding. Our approach for improved big data visual analytics is two-fold, focusing on both visualization and interaction. Given geo-tagged information, we are exploring the benefits of visualizing datasets in the original geospatial domain by utilizing a virtual reality platform. After running proven analytics on the data, we intend to represent the information in a more realistic 3D setting, where analysts can achieve an enhanced situational awareness and rely on familiar perceptions to draw in-depth conclusions on the dataset. In addition, developing a human-computer interface that responds to natural user actions and inputs creates a more intuitive environment. Tasks can be performed to manipulate the dataset and allow users to dive deeper upon request, adhering to desired demands and intentions. Due to the volume and popularity of social media, we developed a 3D tool visualizing Twitter on MIT's campus for analysis. Utilizing emerging technologies of today to create a fully immersive tool that promotes visualization and interaction can help ease the process of understanding and representing big data.

Thesis Supervisor: Vincent W. S. Chan
Title: Professor of EECS and Aeronautics and Astronautics

Thesis Supervisor: Jeremy Kepner
Title: MIT Lincoln Laboratory Fellow

Acknowledgments

This work would not have been possible without all who have helped me along the way.

First and foremost, I would like to thank my advisor Vincent W. S. Chan. His counsel and influence has been a major reason why I have been able to excel here at MIT. Not only has he been able to help me academically, but he also managed to assist me with life's choices as I continued to pursue my career in engineering. He has been patient and always made himself available whenever I needed to address my concerns. I appreciate all of his guidance and am very grateful to have such a close mentor.

Second, I would like to thank MIT Lincoln Laboratory. I can't be more fortunate to have gained a research experience from such a supporting and knowledgable community. MIT's collaboration with the Department of Defense has enabled me to participate in projects with faculty and staff that promote academic, industrial, and governmental pursuits. The endless opportunities this institution has bestowed upon me has laid the foundation of an enriching work experience. To start, I would like to thank Jeremy Kepner. As a respected MIT Lincoln Fellow, he has given advice and support that has carried me throughout my academic career, where I feel more competent and confident in my work and overcoming the challenges yet to come. Next, I would like to thank Vijay Gadepally. His mentorship in and out of the office has led to joint decision making and an improved work ethic that has greatly benefited my time at MIT. Next, I would like to thank Matthew Hubbell. Since my early years as an undergraduate, I have consulted with Matt on many project ideas and topics. He has been my most direct mentor whom I've grown most attached with. Over the years, he has seen my growth where I have honed my technological skills and began to recognize my full potential. I am grateful for all his support and guidance. Also, I would like to acknowledge Albert Reuther. As a pronounced group leader, he was able to effectively convey all that the division has to offer and reflect positively on all that I contributed.

Next, I would like to thank fellow colleagues and interns that have provided me support as I developed this project. Lauren Edwards, for her expertise in D4M[4], pMatlab[16], and Accumulo[2] that permitted integrating new analytics. Taylor Herr for his progress in ingesting and formatting new Twitter data that I have successfully incorporated into the foundation of my thesis. Andrew Uhmeyer for his expertise and insights in game design and animation to help improve the graphics and usability of my project. In addition, Dylan Hutchinson, Kate Thurmer, and many others at Lincoln who have made the workplace a welcoming environment. I am glad to have such endearing friends.

Furthermore, I would like to thank the Unity3D community. Given the privilege to go to Unite Boston 2015[20] this past summer, I attended many talks, demos, and showcases directly related to my research. Unity3D has attracted members from all over the world including artists, indie developers, professors, avid gamers, students, faculty, and leaders in industry to promote a shared passion in gaming and emerging technologies. Their panel of experts provided me with valuable advice that made the final optimizations on my thesis possible.

In addition, I would also like to thank my friends and family for their unconditional love and support. Their constant encouragement helped me continue to remain persistent and diligent in my work when I had my doubts.

Finally, I would like to thank all the financial support I received from MIT to make my undergraduate and graduate academic career possible. Thank you Jim Poitras for your support during my early years at MIT and contributions to Theta Chi Fraternity. I would like to acknowledge Anne Hunter, Vera Sayzew, and the EECS department for all their assistance and direct mentorship. I am very appreciative of the Research Assistantship program; I have been provided valuable support and guidance from reputable associations such as MIT Lincoln Lab, Research Laboratory in Electronics, and CSAIL.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Data is growing ever so fast and requires constant upkeeping. According to a 2011 review by Mckinsey[48], the number of analysts and managers required to fully exploit Big Data analysis is growing rapidly (e.g. approximately 190,000 analysts with “deep-analytical" experience and 1.5 million managers collectively). The desire to gain a sense of intuition on data through analysis is essential to the understanding and promotion of success for one’s business. Determining an underlying structure that best describes the flow of data could uncover hidden connections and patterns that could enhance the knowledge of a network and its users. Many techniques are being utilized to analyze Big Data, however, visualization is one that can very effectively communicate insightful findings.

The gaming industry can be viewed as a medium that has propelled the development of computer graphics and visualization forward. Effective simulations need to incorporate realtime responses and realistic aesthetics to convey meaningful experiences. Games combine both technical prowess and creative ability to produce applications for entertainment, education, training, etc. I have always been an avid gamer and a promoter of gamification. Creating a unique user experience that helps drive insight and discovery is an awe-inspiring pursuit.

My appreciation can be further expressed in my readings of *Ready Player One*[30] and *Ender's Game*[62]. These works realize the concept of immersive environments that can be manipulated by the user's finger tips to complete specific tasks. I am a strong proponent of Human-Computer Interaction (HCI) because this field of study attempts to bridge the gap between people and technology, designing solutions with the user in mind. For example, as users attempt to retain and categorize new information, they add to their cognitive overload by spatially positioning these abstract elements in their head. This can potentially be mitigated through the use of emerging technologies and conveying the same information in a 3D interface that is overlaid in a natural or simulated environment. Tony Stark from *Iron Man 2*[36] and Tom Cruise in *Minority Report*[64] rely on tools utilizing gesture and image recognition to dictate how they want to view their surroundings. Collectively bringing all these digital events into a physical reality is an astonishing feat.

Technologies are growing and performing more efficiently as characterized by Moore's Law[54]. Recently, Virtual Reality (VR) and Augmented Reality (AR) has become a booming industry on the rise these past few years. Ever since the Kickstarter campaign of Oculus Rift in 2012[59], visionaries want to bring virtual experiences to the commercial market and consumers. The goal is to leverage these technologies to enhance and create a realistic environment that further benefits the human condition.

"Seeing is Believing"

Humans rely on perception to aid in their belief that something is real[51]. Visualization and appealing to how one perceives their environment can help enhance situational awareness and decision making skills. If the visual representation is convincing enough, this process can also drive user interaction. An interaction technique is the fusion of all the technological components that represent input and output, and provides a way for the user to accomplish a task[68]. Combining design principles of the user interface with the user experience that better relates to the natural 3D world can yield promising results.

As part of the research and development community at MIT Lincoln Laboratory, I am bringing my insights in Human-Computer Interaction and visualization to attempt to solve a challenging problem. The goal of this thesis is to not only apply visual analytics to Big Data, but to do so in a convincing way that promotes a better understanding of the data network and stimulates user interaction.

1.2 Background

1.2.1 Big Data

“We are in The Age of Big Data”.

Lohr[47] expresses that information is continuing to accumulate and is being collected at an increasing rate. As of 2012, about 2.5 exabytes of data are created each day [61]. Today, big data can be used to convey different concepts such as social media, marketing, financial services, advertising, etc[61]. Much information can be used to characterize particular analytical models in practice; however, this massive intake of information can commonly be unstructured and overly complex. In fact, the main principles that govern Big Data include volume, velocity, variety and veracity[50]. These prime factors make it difficult to easily detect patterns and get an overall sense of the data's architecture.

Volume - unprecedented growth of data intake and storage. Many sources of information exist, resulting in data ingests of massive amounts.

Velocity - speed of data creation and the rate in which it is processed. Determining how data continually flows effects how it can be further monitored.

Variety - diverse, and often unstructured, forms that data acquires. New techniques in organization and representation is needed to simplify complexity.

Veracity - resilience and confidence of data to determine its overall utility. The more consistent the data, the more reliable it is for decision making.

According to Marr[49], another aspect to consider is value. We want to ensure the findings obtained from the analysis are insightful and meaningful. In addition, we want to leverage findings for practical applications. Today's challenges are to develop meaningful tools for analysts and users to understand data in a more convincing way. There are many ways we can attempt to find insight. These often consist of data mining, machine learning, and optimization algorithms that draw in statistics and computer science. Applying visualization is an important technique that's used to effectively communicate, understand, and improve the results of big data analyses.

1.2.2 Visual Analytics

Visualization plays a key role in exploring and understanding large datasets. Visual analytics is the science of analytical reasoning assisted by interactive user interfaces[67]. According to Keim[40], there is much to gain when data is represented in a more visual way. This capability will enable quicker time to insight and more direct interactions with information. Big Data may contain certain anomalies and abstract features that are not so easily recognizable. The goal of performing analytics is to uncover these underlying patterns and display it to the user effectively. This exploration process of Big Data can be improved by integrating human intuition and perception. Hence, the key concept of effective data visualization is to represent congested and complex data in a way that is more manageable for the user.

One strategy is to combine visual analytics with known geographical representations called Geovisual Analytics (GVA). GVA describes the use of visuals with map-based interfaces to further support the understanding of information [38]. The motive for GVA is to get a better sense of large datasets by having a contoured terrain in the background to help guide exploration and analysis. As a result, users gain an additional sense of situational awareness by making comparisons and connections with their surroundings. Geovisual Analytics is also very helpful in determining patterns that may be better depicted when data can be geographically distributed.

1.2.3 Virtual Reality

There have been many approaches of using virtual reality as a visualization platform. VR can be subdivided into a few different techniques. Overall, the goal is to promote full immersion in which a simulated environment surrounds the user. One technique is to have an immersive room with many panels or screens on the walls. Images are projected on these walls, which usually covers all of the user's peripherals. One example of this is "The Cave"[31], which has developed many practical applications. This platform has been utilized for data visualization, geographical exploration, and more gameplay situations. Another less immersive but more focalized form of virtual reality is the responsive workbench[44]. The workbench operates by projecting computer-generated stereoscopic images onto a table seen by a group of users. Users still wear shuttered glasses to get the impression they are viewing objects in 3D. A noticeable drawback, however, is that the simulation's field of view is limited by the sight of the table itself. Most commonly used for VR are Head Mounted Displays (HMDs). This approach provides a stereoscopic display in which two imaging screens are rendered for each eye. Ivan Sutherland created the first virtual reality and augmented reality head mounted displays in the 1960s[65]. However, limitations in processing power and information loss did not make it as usable and applicable during that time. However, advances in CPU and GPU performance have made the virtual reality experience more favorable and sustainable for users. It was not until 1987 when Jaron Lanier coined the term 'Virtual Reality'[46]. Since then, it has been experimented with in many diverse practical applications well into the 21st century, as described in Section 1.3.2.

1.3 Related Work

1.3.1 Social Media and 2D Representations

Social media is a typical use case in the Big Data community due to it's scope and familiarity[26]. It provides a suitable foundation to run sample analyses that can potentially be used to extract more underlying information about the dataset such as overall structure, user behaviour, relationships, trending topics, etc. Massively Parallel Database (MAPD)[10] is one solution for big data querying, visualization and analysis. MAPD is the product of research being done at the Big Data group at MIT CSAIL[13]. With the processing of spatial and Geographic Information Systems (GIS) data, Twitter feeds can be depicted on a large-scale world map. Utilizing a SQL database, the large collection of tweets from this social media can be easily filtered and displayed on a 2D map. This system runs on a hybrid architecture of GPUs and CPUs. MAPD achieves massive parallelism and works well with High Performance Computing (HPC) clusters. This tweetmap represented as a desktop application provides additional functionality such as aggregation filters, collective charts, and query estimations.

TwitterHitter is another Big Data tool that takes advantage of geographic information and geovisual analytics. TwitterHitter [72] is a desktop application developed on the Microsoft .NET framework. This software allows users to access all attributes of available tweets and match them to a user-defined query. The result is then stored on a comprehensive database. TwitterHitter allows users to quickly apply spatial statistics and geographic computational processes on the tweets. The user interface visually outputs the collected results as a linked map, timeline, or a 2D extended graph. This visualization can plot tweets pertaining to a single individual or multiple users. In addition, a live stream view can be activated on the map for real-time analysis. Although MAPD and TwitterHitter are advanced geovisual analytic tools designed for the depiction of large data sets like Twitter, they still do not address the challenge of representing complex multidimensional data.

1.3.2 3D Game Engines and Virtual Reality

When working in spatial and geographical domains, simulations and virtual reality can lead to better discovery. Virtual Reality has made many advances in the realm of game development, most notable for reproducing realistic first person perspectives[69]. Game engines such as Unity3D[19] are capable of constructing user experiences that combine computer graphics, interaction, creativity, etc. all together. They have also been tested for applying techniques such as situational awareness[39] and information visualization[43]. Djorgovski[32] and Donalek[34] have shown how VR has extended from game applications into other areas of research. Some examples of utilizing virtual reality for scientific study include physics[73], medicine[27], and shape perception[74]. These above works have demonstrated how immersion helps scientists more effectively investigate and perceive their area of study. Data visualization has shown to support analyses that are multi-dimensional and highly abstract. According to the MICA experiment[32], utilizing virtual reality helps visualize and analyze large data in 3D space. Caltech[34] shows how VR can create a more collaborative and immersive platform for data visualization. Applying VR technology as a data visualization tool is an emerging field of research with promising outlooks.

1.4 Thesis Overview

Integrating Visual Analytics into Big Data is a challenging problem with many caveats. Our approach is to develop a Unity3D application that takes advantage of geospatial visual analytics of Twitter data at MIT into a virtual reality setting. Although the related social media work of MAPD and TwitterHitter are sufficient Twitter geo-analytical tools, they remain two-dimensional, revealing some limitations in user analytical tasks such as clustering, aggregation, and perception. By embedding catalogued tweets into a 3D geospatial environment, users can more directly perceive and interact with their data. Also, providing a geographical basis can provide additional value and context to the dataset.

The remaining portions of this thesis is structured as follows. Chapter 2 describes the architecture, implementation, and the design of the user interface of our application. Chapter 3 discusses the user interaction our application provides; elaborating on the analytical tasks performed by the user and the narrative this process constructs. We provide a discussion of our results in Chapter 4. Finally, we conclude and mention areas of future work in Chapter 5.

Contributions to the thesis are listed as follows:

- Ingest large datasets that apply high performant analytics
- Visualize data that promotes quicker digestion, ease of manipulation, and further transparency
- Experimenting with virtual reality as an effective workspace and data visualization tool
- Enhance the user experience in a virtual reality platform

Finally, the appendix lists additional tables, figures, and references that complement the material in this thesis.

Chapter 2

Application

2.1 Architecture Overview

This thesis has integrated many core technologies that have been in development these recent years. We attempted to combine these commercial technologies with innovative applications and analytical models developed at MIT Lincoln Laboratory.

2.1.1 Technologies

When planning this project and conceptualizing its design, we fully considered which emerging technologies and hardware we wanted to utilize. For rapid prototyping, we preferred equipment that was commercially distributed, readily available, and provides reliable developer support. We were more inclined to use inexpensive commodity software\hardware to aid in the development process. After review, we decided to use the Unity3D Engine with the Oculus Rift headset and Leap Motion controller. Given the software developments kits (SDKs) with Unity3D integration, we can build an application that can run on the traditional laptop computer. These devices would provide the foundation for an immersive and interactive data visualization analytical tool utilizing a virtual reality platform. Even though most of these technologies are new with much improvement still to be made for development, they are sufficient for research and have promising outlooks.

2.1.1.1 Unity3D Game Engine

This project embedded information from large datasets into the *Unity3D*$^{\text{TM}}$ game engine[19]. Many reasons demonstrate why Unity3D was the most reliable game engine to develop on for this research. Unity3D is a fully capable physics engine that is highly reputable in performance. Many features are readily available for developers at varying subscriptions. Its flexibility in multi-platform support and scripting makes it a valid candidate as a modeling and 3D visualization tool. Unity is built off the .NET framework, where programers can script game components in a 3D scene using object oriented programming. As described in Section 1.3.2, Unity3D is extending its capabilities as an effective visualization tool into markets outside of the gaming industry, such as research and academia. It is also becoming one of the leading development tools for virtual and augmented reality.

2.1.1.2 Oculus Rift

Head Mounted Displays (HMDs) are wearable devices placed on the head with a display covering the eyes. This optic is typically stereoscopic where an image is rendered for each eye. The screen's placement and orientation relative to the eyes can mask the majority of the user's peripherals. This allows the projections and images shown on the display to be fully immersive. Many key factors are considered to describe the performance of HMDs. Interpupillary Distance (IPD) measures the distance between the pupils, which is necessary for determining focus and the overlapping viewing areas. Field of View (FOV) is the extent of the environment that is observed. Humans typically have about 180 degrees FOV. Varying the field of view for HMDs will effect the immersion felt by the user. Resolution of the display specifies the pixel density. Given the display is already in close proximity to the face, a higher resolution is preferred to allow for better quality and more realistic simulations.

The Rift is a virtual reality HMD developed by Oculus VR[15]. Since 2012, the company has been developing the Oculus Rift to be a leading platform for virtual reality. The device is produced as a secondary display that is tethered to a personal computer for processing. They have also made efforts for mobile versions such as the Samsung Gear VR[17]. For developers, Oculus has released two development kits. The initial DK1 includes a gyroscope, accelerometer, and magnetometer for improved rotational tracking. However, limitations such as latency and low resolution were noticeable and detracted from long term gameplay. The second version revealed in the DK2 has better screen resolution, reduced latency, and a higher frame rate. The OLED display has a HD resolution with 1080x1200 pixels per eye and a refresh rate of 90 Hz. It also has a FOV of roughly 100 degrees. A supplemental Infrared (IR) sensor aimed at the front of the Rift adds three axiis of freedom for positional tracking that more accurately monitors head movement. Throughout the duration of this thesis, we have prototyped on both versions and updated assets in the project accordingly.

2.1.1.3 Leap Motion

Despite conventional I/O for desktop displays such as the mouse and keyboard, supplemental devices are necessary to help relay user input in VR. When using a HMD that simulates a 3D setting, the player no longer has full awareness of his or her physical world in reality. Naturally, the player relies on their senses and interaction with his or her surroundings in order to correctly navigate the scene and dictate action. Given this telepresence of "feeling like you are there", additional mediums that are in accordance to the player's simulated environment is much more desired.

"3D Output Meets 3D Input"

The Leap Motion[9] controller attempts to address these concerns. First launched in 2013, the Leap Motion controller is a USB device designed for hand detection and gesture recognition. An image is generated from each of the two monochromatic IR cameras in the device, representing the live feed of a black and white "speckle pattern" from the forward-facing infrared LEDs. Using machine vision and applied depth-mapping algorithms, correspondences can be distinguished from the 2D images and 3D positional data can be synthesized. Leap Motion focuses primarily on detecting the body parts within your hand (e.g. palms and fingers), allowing for more fast and accurate hand tracking. The Leap Motion's solution for hand tracking is analogous to the Microsoft's Kinect[12] for body tracking. Depending on the hardware, the controller can reach up to approximately 200 FPS. The interaction zone in which hands are most precisely tracked is about eight cubic feet (and one meter in the camera's forward direction) with a Field of View of 135-degrees. This is complementary to the Rift's FOV mentioned in Section 2.1.1.2. Leap has already begun their journey creating virtual reality applications and have released easy-to-install VR developer mounts for HMDs.

2.1.2 Integration

Our goal is to integrate all of these devices into one application to be used for data visualization and analysis. Development was completed on a Mid 2009 15-inch Macbook Pro. All sensory inputs from cameras and USB devices were read directly into the computer with a powered accessory USB hub. The application binary can directly output to the Rift display via HDMI. For testing purposes, we still incorporated traditional input devices such as a mouse and keyboard, but also introduced the XBOX 360 controller for user navigation and selection. For a full list of equipment, software, and hardware specifications, please refer to Appendix A.

An overview of the overall integration of these technologies and a diagram summarizing the architecture of this project can be seen in Figure 2-1. The components mainly consist of (1) Data Extraction, (2) User Input, (3) Game Engine, and (4) Visualization Output. Each module is described as follows:

1. **Data Extraction** Data is provided and collected primarily from MIT Lincoln Laboratory's database. These are in the form of parsable TSV files or portable FBX models. Open source file readers and 3D software applications are used to further customize the formatting and representation of this data to be used in the gameplay application. Section 2.2.1 provides a detailed description on how

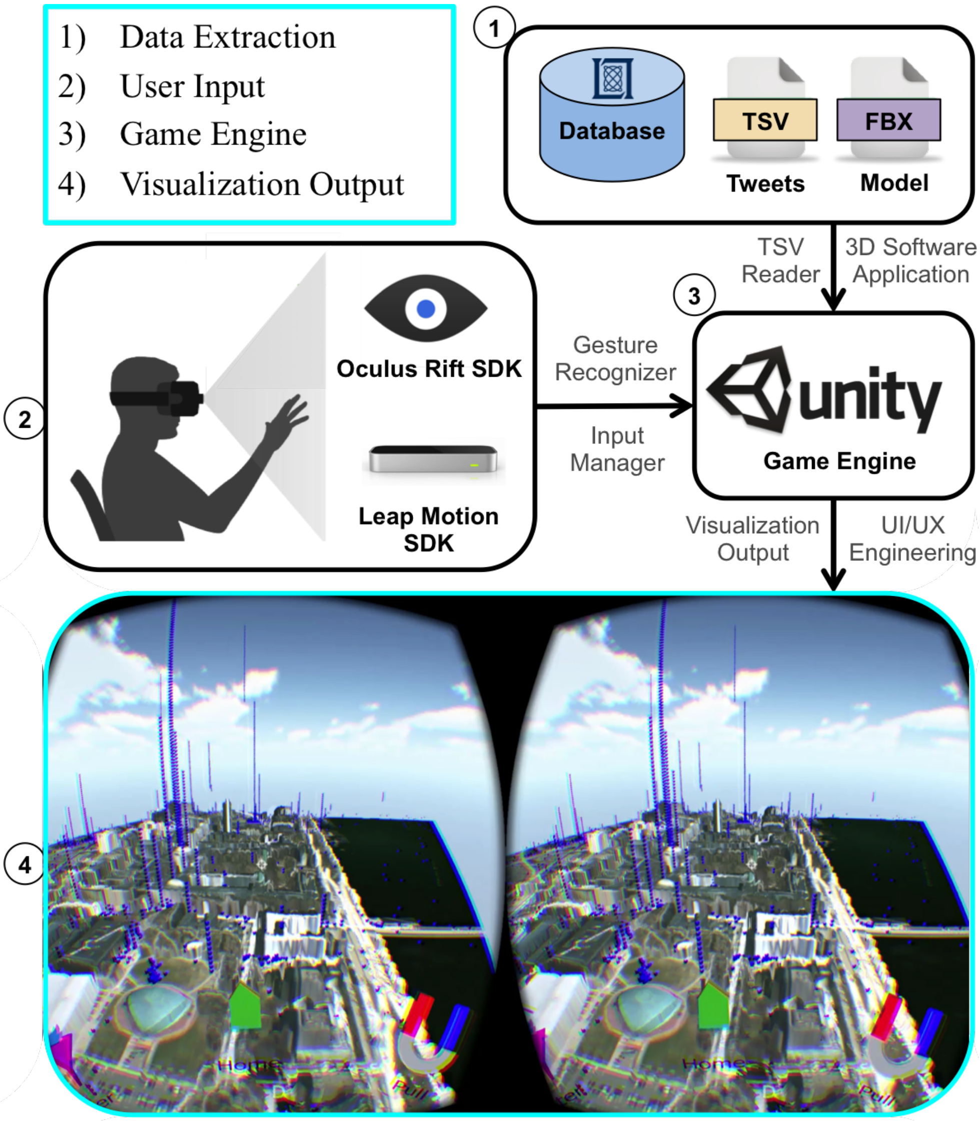


Figure 2-1: Overview of Application Architecture
Four main components drive the application; (1) Data is first extracted as parsable files from MITLL, (2) User input devices such as the Oculus Rift and Leap Motion are used to control player movement, (3) Unity3D game engine is used for hardware integration and software development, lastly the (4) Visualization is represented as a 3D rendition of MIT's campus enriched with Twitter data and interactive user elements.

information has been obtained and pre-processed for both static and dynamic data.

2. **User Input** As expressed in Section 2.1.1, modern equipment is needed in order to correctly monitor user input in a virtual reality setting. The processing power to record sensory information, camera orientation, positional tracking,

etc. requires the utmost speed and accuracy to convey a convincing and smooth simulation. Scripts posing as managers keep track of player status, recognized gestures, and user-defined actions in order to guide the proper game response.

3. **Game Engine** Unity3D is the game engine we utilized for core development. Given that the Software Development Kits (SDKs) used are open source and easily integrable into Unity3D, communicating between devices is more seamless. We can now exploit the gameplay capabilities of Unity3D to produce an effective interface design and create a distinct user experience.

4. **Visualization Output** During gameplay, the HMD displays a stereographic view from the player's perspective. With correct camera placement and high quality resolution, the visualization is meant to be realistic and aesthetically pleasing to the naked eye. More information concerning the game mechanics and output generation are discussed in Section 2.2.2.2.

2.2 Implementation

2.2.1 Data Extraction

Developing an accurate geographical environment into a 3D simulation is important for effective situational awareness and user analysis. Data sources can often be inconsistent and diverse; therefore, much pre-processing is involved to ensure optimal data is used for visualization and scene creation. In the next subsections, we will describe two key sources for our data, LADAR and Twitter, and how they have been further customized to provide as the foundational basis of this project.

2.2.1.1 LADAR Data

As a sensing technology developed at MIT Lincoln Laboratory, LADAR is utilized to generate 3D representations of global locations[28]. LADAR measures the distance of reflected light from a laser source to an illuminated target as an accurate metric for height mapping. In 2005, a LADAR dataset was collected from an overhead aircraft over Cambridge, MA encompassing MIT's campus[37]. With about 1m resolution, a dense height map was created where each planar point corresponds to the altitude at that location. The final image resulted in a 1.0km x 0.56km region of Cambridge, as shown in Figure 2-2.

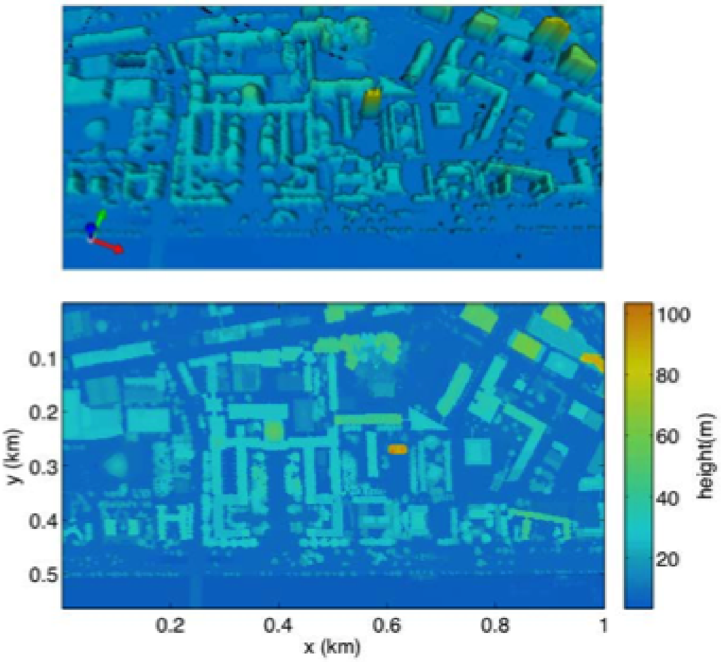


Figure 2-2: LADAR Image of Cambridge
Height map of Massachusettes Institute of Technology generated from LADAR data in 2005. Final scan shows a 1m resolution image of approximately one square kilometer of Cambridge, MA.

To produce a 3D model of this particular region, the LADAR data is converted into a stereolithography STL file. This is a common 3D file format that can be imported to various modeling programs for further customization and enhancement. 3D graphics and animation software such as *Blender*$^{\text{TM}}$[3] and *Maya*$^{\text{TM}}$[11] was used to aid in optimizing the view of campus. For noise reduction, mesh smoothing algorithms was used to smooth jagged vertices and get rid of any outliers. These were than exported to a FBX format so that it can be read into Unity3D.

Given this region of Cambridge, satellite imagery from Google Earth[7] provides additional context of the setting. The longitudinal and latitudinal bounds of the area is (+42.3638, -71.0812) to (+42.3557, -71.1032). Two square JPEG images, corresponding to about roughly one half km in world dimensions, were extracted from Google Earth to capture the entire scene. These were then compressed into textures, each 2048 x 2048 pixels. Section 2.2.2.2 describes how these images were later used in the construction of the game environment.

2.2.1.2 Twitter Data

The Big Data source on which we wanted to perform further analysis is Twitter. Twitter is a social media blogging site where users can post messages in the form of tweets of 140 characters or less[45]. If posted from a mobile device, tweets are bound with a geo-tagged location in addition to their username, text message, timestamp, etc. One of the key challenges associated with the research on Twitter data is in the searching, aggregation, extraction, and analysis of a large collection of posts. Analyzing tweets can help provide insight on many events such as social behaviours, controversial topics, user reputation, and popular locations.

Initially, these tweets are gathered from Twitter Decahose[5], which provides 10% of all random tweets, and can be narrowed down to user-defined criteria (e.g time and location). In 2013, an MIT CSAIL[13] initiative collected geo tweets on MIT's campus known as the Twitter Corpus[66]. This data originally spanned approximately three months from April 2013 to July 2013 and contained about 450 million tweets. Since then, MIT Lincoln Laboratory was able to continue retrieving and ingesting this data on a database to run additional analytical models. Results were exported to a readable TSV format as described in the following Section 2.2.2.2.

In our first visualization, we extracted about 6,000 tweets over the course of five months from October 2013 to February 2014. As of 2012, Twitter has announced a powerful open source API that permitted a full history of tweets[6]. This allows more freedom to explore the Twitter dataset with additional user-defined criteria and an expansive search history (as opposed to the original limitation of only three weeks). Internally, MIT Lincoln Laboratory conducted an open source data initiative to collect and archive live tweets. As a result, we were able to update our collection of tweets to display roughly 10,000 tweets from January 01, 2015 to July 25, 2015.

2.2.2 Pre-Processing

After ingesting the raw data, it is parsed into a tab separated value (TSV) format and stored on the high performance database (DB) Apache Accumulo[2]. Using the same procedures exercised by Weber and Gadepally[70], additional models can be used to further query the data. Specifically, we utilized the Dynamic Distributed Dimensional Data Model (D4M)[42], a high performance schema that can be used with Accumulo. This permitted a customized pipeline to refactor the data described below in Section 2.2.2.1.

Additional configuration went into initializing the game scene and generating 3D models. Given geographical data, global and local normalizations were needed to accurately depict MIT's campus. Visual efforts to efficiently represent Twitter data aided in creating a more convincing environment. We also calibrated the player and camera during instantiation to more directly correspond to user inputs and desired actions.

2.2.2.1 Data Pipeline

We propose a pipeline that, given a collection of raw data, a researcher can perform analytics on a subset of interest. Shown in Figure 2-3, we can generalize the pipeline as it pertains to the D4M model. This can be described as (1) Parse raw data into triples to be inserted into the database, (2) Ingest triples into the database, (3) Query graphs from the database, (4) Analyze graphs using analytics and other methods.

1. **Parse** After collecting the raw Twitter data, the tab separated value (TSV) file format is parsed to construct Associative Arrays. Associative Arrays represent complex relationships of data either in a sparse matrix or graph form[42]. More information can be found in Appendix C.1. Each parsed file creates three additional files pertaining to the triple (row, column, value) store. We reflected this in the Twitter data; each row pertains to the tweet ID, each column pertains

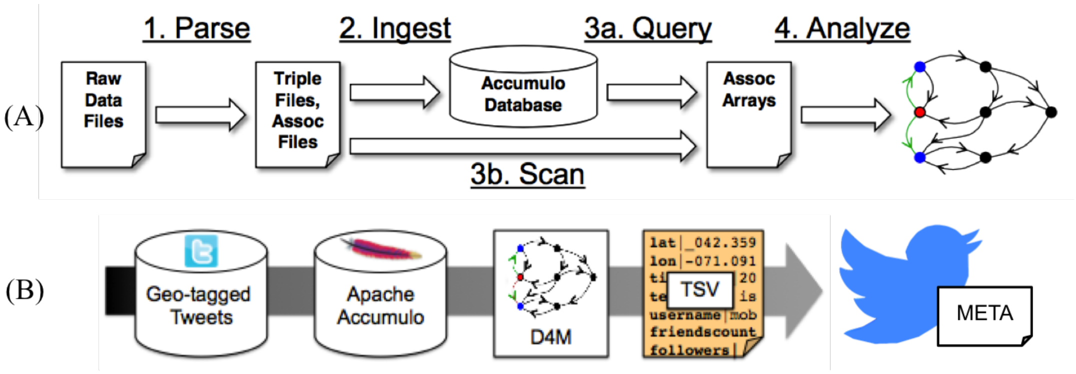


Figure 2-3: Data Pipeline for Twitter Analysis
(A) Generalized data pipeline in which raw data files are represented as row, column, value triples and ingested on a database to be queried and analyzed using D4M. (B) Geo-tagged tweets are stored and pre-processed into a TSV format which can be parsed to render 3D objects in the scene.

to an attribute of a tweet (e.g status, username, location, etc), and each value is the original tweet given a row,column pair.

2. **Ingest** Data is ingested onto the Accumulo database as four main tables as shown in Figure C-2. Tedge shows a relationship between a tweet ID and a particular entity, whose value it represented as a boolean for the row,column pair. TedegDeg is the sum of column,value pairs. TedgeTxt shows the original tweet text. This representation allows for D4M to be performed in the following query step in the pipeline. The total time to ingest all the data took about 14 minutes. Figure 4-3 in Section 4.2.1 graphs the time it took to ingest the data and comments on how parallelism improves performance.

3. **Query** Now that the data is ingested and parsed in the Accumulo database, it is possible to query using D4M. D4M is an innovative new programming model that combines numerous processing techniques such as Linear Algebra, Associative Arrays, and Triple-Store databases. The D4M syntax allows for easy data filtering by latitude and longitude, as well as quickly inserting additional attributes to tweets that satisfy certain criteria. Table 2.1 shows some types of queries performed using the simple format of Associative Arrays. Additional queries can be beneficial for producing further filters and analytics when con-

Table 2.1: Analytical Functions Performed on Associative Array

Type	Sentiment	Language	Links	Hashtags	Periculum	Bounds
Row	'word_lower'	'lang'	'word'	'word'	'word_lower'	['lat', 'lon']
Col	'score'	'lang'	'links'	'hashtags'	'periculi'	['lat', 'lon']
Params	source = ... 'AFFIN-111.txt'	langs = ['en']	links = ... ['http', ... 'https']	hashtag = '#'	p = ... ['danger', ... 'stranger', ... 'evil']	bounds = ... [42.3557, ... -71.0812, ... 42.3638, ... -71.1032]
Filter	See Figure C-3	LANG = A(:,CatStr(’lang|’,langs));	LINKS = CatStr(’word|’,links);	None	PERI = CatStr(’word_lower|’,p);	lons = [’lon|-71.1032,:, lon|-71.0812,’]; lats = [’lat|42.3557,:, lat|42.3638,’];

Extraction	Aout =
	DetermineScore(A);
		Aout =
		A(Row(LANG),:);
			Aout =
			A(:,StartsWith(
			LINKS));
				Aout =
				A(:,StartsWith(
				’word|#’));
					Aout =
					A(:,StartsWith(
					PERI));
						LON =
						A(:,lons);
						A =
						A(Row(LON),:);
						LAT =
						A(:,lats);
						Aout =
						A(Row(LAT),:);

figuring the simulation.

4. **Analyze** The final step is to perform analytics on the developed graph. This is highly dependent on the goals of the researcher. As an example on the Twitter dataset, we can perform sentiment analysis and attempt to gain an overall sentiment of a tweet based on the words within the original post. We used sources such as Matlab[63] as reference that utilizes a sentiment dictionary and creates a summation based on a score for each word in a post. Figure C-3 shows the simple function called to append a score value on the Associative Array. Figure C-4 shows a visual representation emphasizing the benefits of representing these relationships as sparse matrices where linear algebra can easily manipulate and customize these arrays.

Post-processing, additional tasks can be performed. In particular, we wanted to perform in-game tasks representative of what an analyst would like to explore and research on a large data set. After the queries and analytics are performed above, we can embed this as a TSV file that will be associated as meta data in the visualization. This allows additional tasks such as dynamic filtering, aggregation, and adjustable zooming to be done at runtime. These are later described in Chapter 3.

2.2.2.2 Model and Scene Formation

Creating the static scene requires some manual configuration. The textures provided from Google Earth[7] were rendered on scaled 2D planes placed at the scene's origin. Importing the raw LADAR FBX model into Unity produced several model subdivisions. One constraint of Unity is that each imported model is limited to 65,000 vertices before partitioning itself into new models. These models were arbitrarily sectioned and not necessarily positioned relative to the game's point of origin. A global rotation and translation in the scene was performed on each section to properly connect models and ensure their positions matched correctly on the ground texture. Similar to Google Earth and LADAR data collection, Unity's default unit is 1m. This made transitioning and manipulating elements in the scene very consistent.

Tweets additionally needed to be represented in the 3D world. An open source delimited file reader was used to parse each tweet as an individual record, given a pre-defined header. Of all the tweets, approximately 98% were read in fully. Ambiguous tweets that included unrecognized characters, invalid values, and/or missing fields were ignored. Translating these records into the game environment required the use of publicly available models provided from Google Sketchup 3D Warehouse[1]. *Maya*™[11] was used for provide further model enhancement and customization. Attributes of each record coordinated which 3D model to use. Figure B-1 shows an example of how tweets are shown as blue birds by default whereas those containing the word “danger" are represented as red skulls. This corresponds to the result of a string matching analytic performed by D4M, as previously described in subsubsection 2.2.2.1.

Additional work was required to correctly map the geographical information provided by a tweet into the game world. From Section 2.2.1.1, the latitude and longitude boundaries of the LADAR and Google Earth images are well defined. Therefore, translating real world latitude, longitude locations to game coordinates required a simple geometric transformation onto the scene's game ground layer. Figure 2-4 shows the final 3D rendition of MIT's campus after all the necessary transformations have been completed.

Configuration also needs to be completed for the player. Initially, the user is instantiated as a first person controller. With a free-form camera, the player's perspective can dynamically change in the x,y,z directions and is free to navigate within the bounds of the scene. Colliders on buildings, tweets, and other 3D models prevent the player from reaching areas with obstructed views within objects. Script managers have been assigned to keep track of player state when they navigate and interact within the scene. Caching the player's current view and move directions helps dictate which player actions are currently permitted. Additionally, many efforts have been made to construct a convincing user interface that promotes further exploration on the dataset.

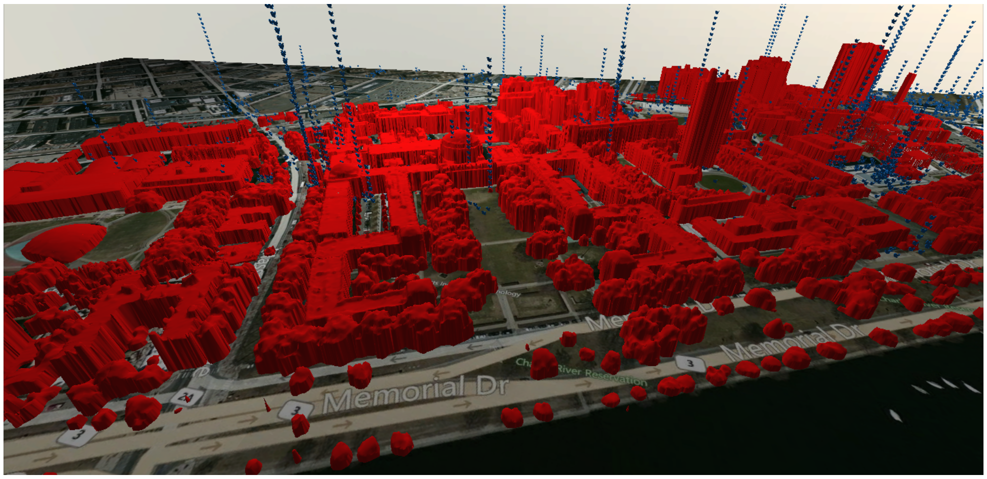


Figure 2-4: Rendition of MIT's campus imported FBX model
View of imported MIT FBX model into the Unity3D engine as seen from game's free-form camera. These models are then superimposed on Google Maps textures matching the same scale and latitude, longitude bounds as the original LADAR data. Tweets are juxtaposed onto the scene based on provided mobile geo-tagged information.

2.2.3 User Interface and Design

With the static scene configured, additional elements are implemented in the environment to enhance immersive gameplay and promote visual analytics. After a player has been instantiated, the design of the user interface dictates how the user will interact and how effective these related affordances convey player intentions.

2.2.3.1 DK1 Prototyping (And Lessons Learned)

In the first iteration, we developed and designed with the DK1. Once able to construct a 3D scene, it was now time to transform the player experience into a first person perspective. Below we mention and evaluate preliminary developments on the user interface design.

Standardizing input for virtual reality poses as a challenging problem. The original development kit for the Rift did not provide an out of the box solution so we had to rely on typical input devices such as the keyboard, mouse, and the XBOX 360 controller for user interaction. When 'wired in', it's difficult for a player to use devices that are not directly simulated and perceived in the environment around them. The gamepad controller proved to be the most prominent exception due to its familiarity in mainstream gaming and its minimal button and joystick layouts that made it easier for user recall.

To confirm player direction and orientation, a 3D cursor/crosshair is shown on a transparent texture in front of the player's camera. This is used to also help pinpoint where on the 3D scene the player is currently looking and facing. As the player is constantly moving, the cursor remains in the center of the screen. If the user chooses to pause player movement, the cursor is no longer fixed and is free to interact with game elements within the camera's current field of view. Movement of the crosshair is then mapped by the joystick on the gamepad controller with selection events enabled by a custom input module.

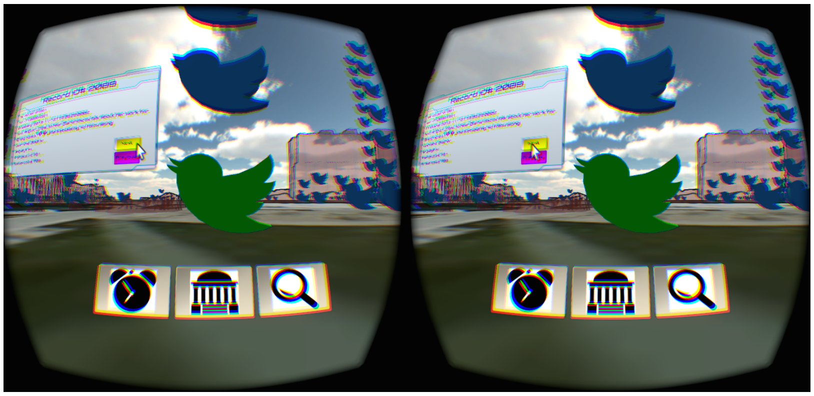


Figure 2-5: Mockup of GUI
View of selected tweet and HUD as seen from the stereoscopic view of the Oculus Rift, a VR device described in Section 2.1.1.2. Utilization of 3D space allows freedom of GUI placement; whether at a fixed distance in front of the player or on 3D objects

As shown in Figure 2-5, additional GUI elements on the Heads Up Display (HUD) are displayed to help guide the player into further investigation on the Twitter dataset. We incorporated a dock similar to the Mac OSX[53] that was at a fixed distance from the player's camera, oriented at the bottom of the screen. Introducing a dock taskbar with icons was to invoke familiarity to the user. Showing users items that are easily recognizable improves usability over needing to recall items from scratch because the extra context helps users retrieve information from memory[24]. These icons convey options users can perform on the Twitter dataset such as filtering time ranges, changing object opacities, and executing string searches. These analytical tasks and enhanced user experience are described in Chapter 3.

Other interactable elements can be displayed and projected onto the scene itself. The meta data associated with the tweet are represented as 3D displays analogous of a speech bubble popup. This reveals all the original data as it was read in such as username, follower count, timestamp, text, etc. However, there were some limitations when viewing text at a far distance. Due to a low resolution, any readable text and menus required to be positioned close to the user. Pixelation added stress on the eyes and made aliasing more prominent. Also, occluding the view of the player interrupted the connection with the environment and often led to a noticeable break of immersion.

Another input device we utilized to help promote interaction with the dataset is the Leap Motion controller. Tests have been done to create a hybrid solution where you can use a controller with one hand and have the Leap Motion recognize the other hand. Hand detection and recognition is highly dependent on CPU performance and framerate. At times, the hand can often not be fully recognized if portions of the hand are being occluded or not oriented properly. This became difficult when attempting to use raycasting as a selection event in the scene. Naturally, the user is not completely stationary and small deviations in hand movement resulted in a large margin of error during gameplay. Figure 2-6 illustrates how a user's hand is rendered in the gameplay environment.

Developing with the Leap Motion and XBOX controllers had some limitations on the DK1. Many affordances still referred back to 2D conventions such as mouse selections and displaying multiple screens. Also, some events were tediousness and less forgiving such as moving a cursor with a joystick and raycasting with the Leap Motion controller. The next section describes another iteration for a user interface with the ergonomics of the user in mind. Furthermore, improvements on software and hardware specifications on the DK2 helped promote immersion and interaction for the player.

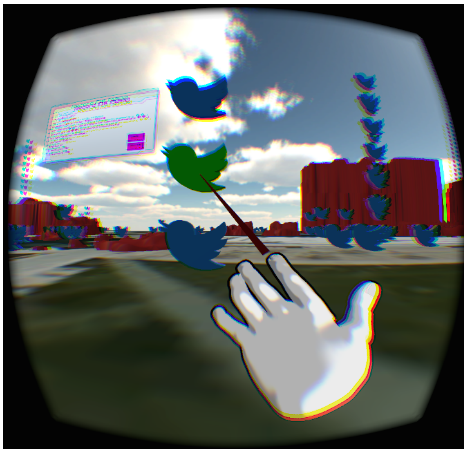


Figure 2-6: Debugging with the Leap Motion Controller
Leap Motion hand controller allows the player's hands to be rendered in the simulated scene. Gestures and other inputs registered by the device can launch events and other commands intended by the user during analysis and gameplay.

2.2.3.2 DK2 and an Improved Holodeck

The Rift's introduction of the DK2 has brought many improvements that benefit player interaction. Reduced latency and a higher resolution screen gives the user a more realistic viewing experience with enhanced input tracking. Head and positional tracking provides a more comforting experience as the view direction and movement is synchronized with the user's real-world pose. As mentioned in Section 2.1.1.3, the introduction of the Leap Motion controller can now incorporate input from the physical world, most specifically the hands, into the simulated digital world. However, the interface needs to be configured carefully to stimulate 3D interaction and prevent any flaws that lead to an unconvincing setting. Below is the design process of creating an enhanced user interface that creates a more natural work station environment for task management.

Affordances refers to the physical characteristics of an object that guide the user into using that object[57]. Therefore, the representation of an object is an indicator in which how the user plans to undergoe interaction. Design principles are necessary in making sure these affordances remain effective and convincing. In particular, the more physical the response of an object, the wider the range of interactions that may correctly correspond to that object. Making these elements always respond to player action additionally creates a sense of realism.

With the Leap Motion controller, we can combine these affordances with responsive gestures. However, one thing to consider is that there is no haptic feedback when interacting with digital content. Therefore, we have to rely on depth cues to simulate tangibility. Objects that appear farther away should have less contrast and appear less sharp than those that are in close vicinity. Lighting and shadows can be a supplemental reinforcement. For example, when a hand approaches a button, the orientation of the occluding fingers should effect how shadows are casted. Sound also plays a critical role in user confirmation. Audio feedback, such as clicking, reassures a player that an action has been completed. These indicators have been taken into deep consideration in the UI elements of the workspace.

When simulating the workspace, we want to take into consideration the ergonomics of the user. Designing an interface based on how the human body works can lead to more "intuitive" interaction that seems more natural to the player. For example, when in a seated position, more strain is placed on the neck when a player attempts to adjust their view to look behind. Also, hands and arms tend to move in arc-like shapes rather than straight lines. Alger[22] emphasizes the importance of placeholders to illustrate where UI content should be positioned when promoting productivity in VR. For readability, UIs should be a 3D part of the virtual world within three meters from the user. They should also respond accurately to player head movements. As shown in studies conducted by Chu[29], there are constraints on the maximum range and angles in which digital content can be viewed by a user in a virtual reality setting.

Our dock has attempted to utilize all these features in our first iteration. A 'hoola-hoop' design with widgets surrounding a player as seen in designs by Abovitz[21] and Leap Motion[55] seem to be an effective UI for a 3D task management structure. This can be seen as a "cockpit" that surrounds the player. This design is fixed onto the player, which helps provide additional reorientation during player movement. Our efforts were to incorporate a generalized operating system tailored for scalability and usability. Figure 2-7 shows the overall schematic of the cockpit interface. In summary, there are three main components:

Banner Radial strip above the player. Non-interactive and relays information to the user such as time, direction, objective, etc.

Screen Space surrounding the player at eye height and composed of interchangeable panes. Each pane is divided into three parts. The middle is unoccluded and left open for the player to continue viewing the game environment. The left and right sides are synonymous to one another. These regions are intended for cascading panels that resemble traditional 2D displays. The main component of a panel is a menu that displays content. The remaining portion is a control which drives the content with interactable buttons or informational text.

Dock Radial taskbar situated at desk height slightly beneath the player. The dock is composed of evenly spaced widgets portraying 3D buttons corresponding to each pane. When a widget is selected, a pane is activated revealing all of its child elements.

The progression of the dock design can be viewed in Figure 2-8 and Figure 2-9. It attempts to incorporate key features when browsing in 3D space. Allowing more space allows for more productivity according to Leap Motion's Blog[52]. Being able to have a physical arrangement of display windows makes it easier for the player to have an organized space. This arrangement also leads to a reduced cognitive overload. With spatially arranged interactable elements, the player can readily recognize where to perform desired actions; leaving more room for working memory. This correlates directly with dimensionality. Humans can tell when objects are on different viewing levels, decreasing the time to inherently figure out the associated depths.

Additional design choices have been made to help the player select in the simulation. Using procedures shown in Oculus's Documentation[14], we can customize the reticle to adjust in scale and position relative to the player's view direction. By drawing the crosshair at the same depth as the object it is targeting, the doubled image 'cross-eyed' effect is removed (which usually happens when the eyes converge on a plane not at the same depth as the object). In addition, it was necessary to customize the input system module for the Leap Motion controller. Player states during gameplay needed to reflect whether the user action is enabled and can interact with the scene.

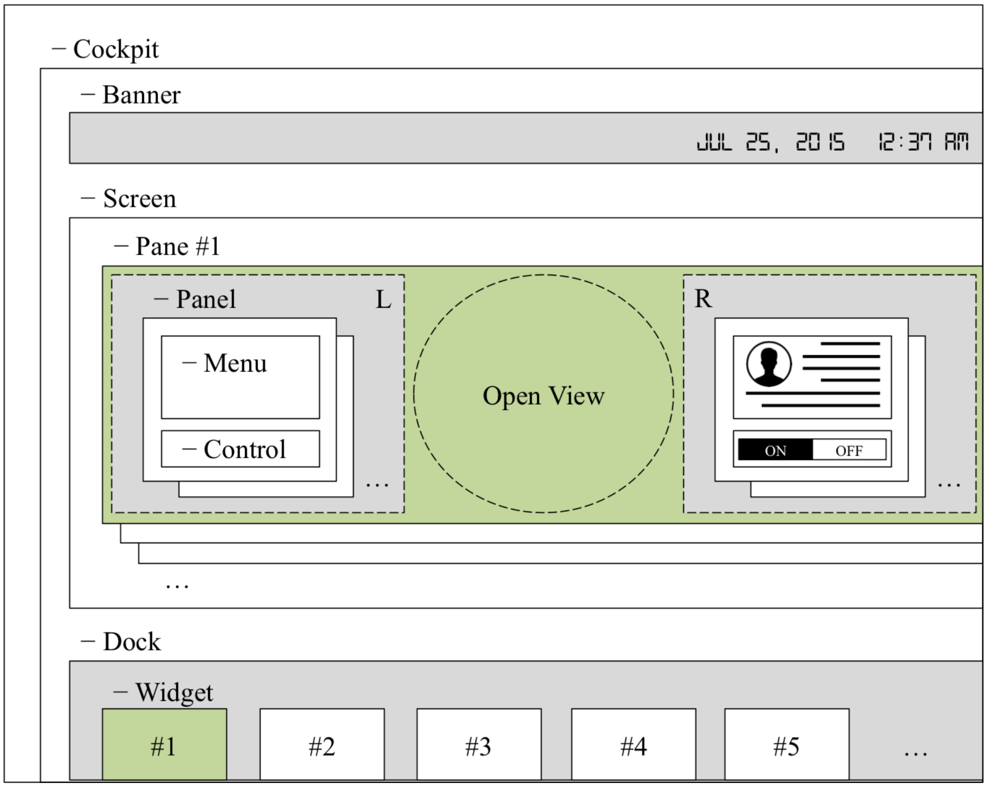


Figure 2-7: Schematic of VRLeapInterface Holodeck
2D representation of the user interface task management system. Composed of three main elements: (1) Banner - above eye height, (2) Screen - situated at eye height, and (3) Dock - located at desk height.

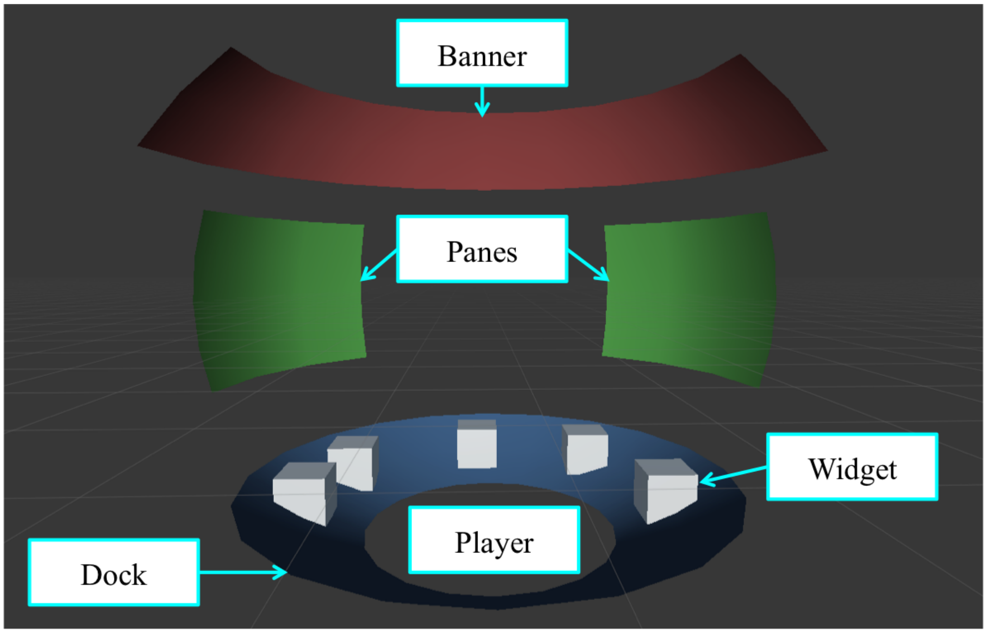
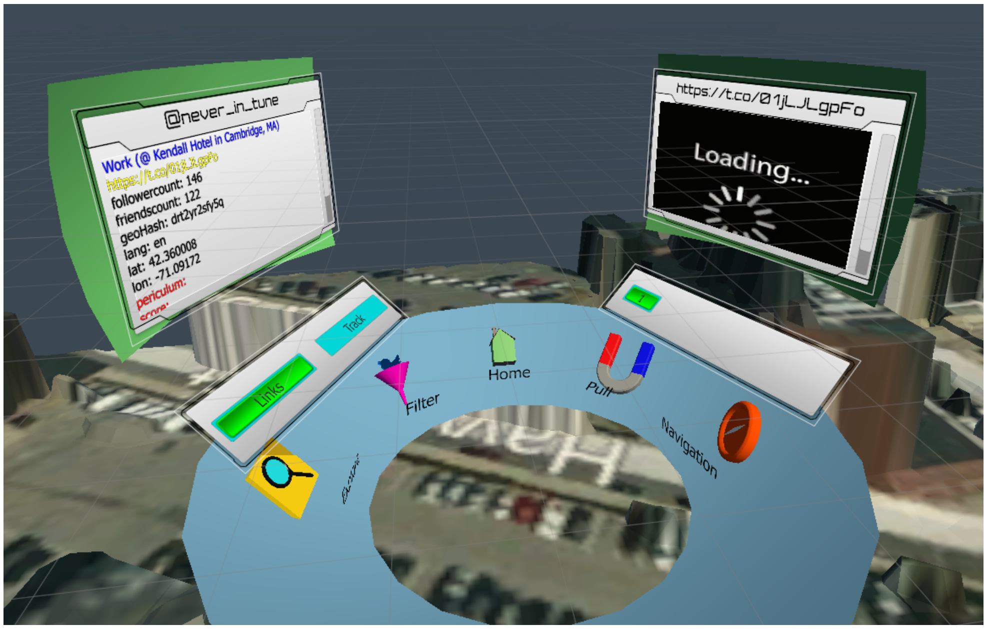


Figure 2-8: 3D mockup of VRLeapInterface Holodeck
Early stages in prototyping of the 3D user interface. Placeholders are constructed using the software Maya to illustrate optimal views and locations of displays and interactable elements.



@never_in_tune

Work (@ Kendall Hotel in Cambridge, MA)
https://t.co/01jLJLgpFo
followercount: 146
friendscount: 122
geoHash: drt2yr2sfy5q
lang: en
lat: 42.360008
lon: -71.09172
periculum:
score:

https://t.co/01jLJLgpFo

Loading...

Figure 2-9: Iteration of VRLeapInterface Holodeck
Iterating on the mockup which originally had placeholder elements, components are now implemented and connected with one another in the user interface. Specific user actions dictate what state and items are visible in the UI.

Chapter 3

User Experience and Interaction

3.1 Analytical Tasks

Interaction techniques fuse together user input with output to provide a better way for a user to perform a task[60]. Common tasks that allow users to gain a better understanding of data include scalable zooms, dynamic filtering, and annotation. Below, we describe some tasks pertaining to data analysis that can be performed fluidly by the user in our application of the MIT Twitter dataset.

3.1.1 Navigation and Exploration

Creating a life-sized simulated setting enables the player to naturally move about the scene. Virtual reality fully immerses the player and enables a constant stimuli for exploration and discovery. Using MIT's campus allows players to recognize familiar landmarks and discover new Regions of Interest (ROI). The utilization of a free-form camera permits different perspectives that would not have been so credible in the real world. Adjustable zooming is possible by having the camera move closer or farther from a relative position in the scene. The user's freedom to move about the 3D scene is key to revealing the overall framework and features of the dataset which would not have been so noticeable on a traditional display.

Locomotion, however, remains an issue within VR due to effects of simulator sickness. Our use of the gamepad controller can effectively move the player in the scene along the planar x,y,and z directions. However, if the user's look direction does not necessarily match the move direction, this discrepancy can create a sense of nausea. Augmenting the cockpit with a navigation widget that contains a list view of locations helped solve issues pertaining to disorientation in the landscape and appealed to the desire of "jumping" to locations. Figure 3-1 is an example of this 3D element as seen by the player. When the user selects a Point of Interest, they are teleported using a linear interpolation between the player's current position and the selected destination. The transition is initially fast but gets smoothed and slows as the destination is reached. Exploiting the fact we are using a 3D game engine allows for quick physics calculations and collision detection if necessary during player movement.

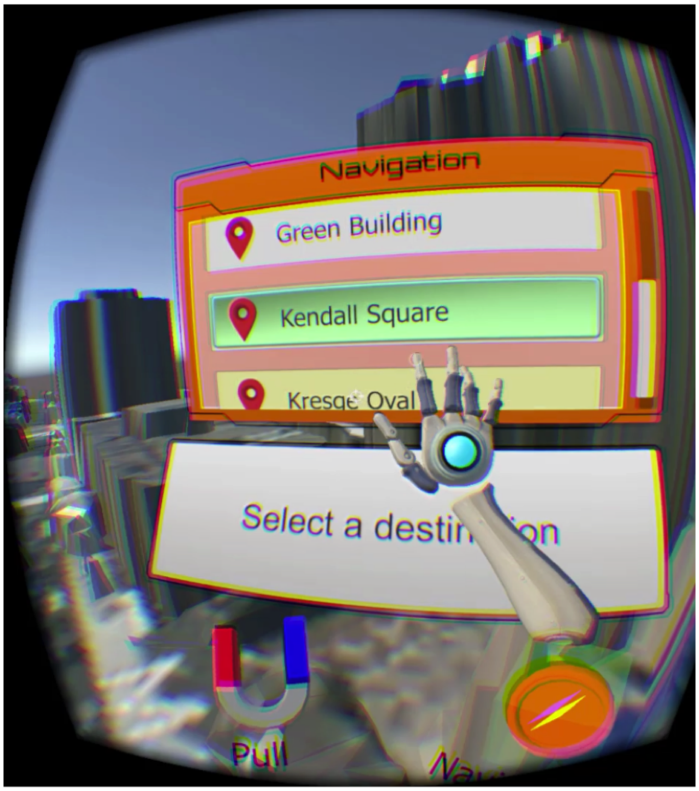


Figure 3-1: List View of Teleporting Points of Interest
During gameplay, the player can navigate by selecting from a dropdown of nearby locations or Points of Interest. Visual queues such as color contrast and 3D movement provides the player useful feedback while using this virtual user interface.

When a tweet mentions a location, the post provides additional context and valuable information that could determine a defining characteristic of an area. Additionally, directly navigating to that area can give more insight on how a collection of tweets are gathered or the overall significance of that Point of Interest. For example, many people tweeting in the Kendall area post about “going to work" or “grabbing food". Navigating to that portion of the scene, we can see that this is a common place for commuters; being situated at a T station with local shops near by. We can also explore an area in which a user may not necessarily have any prior knowledge about. Recognizing a collection of tweets in a certain area and extracting common phrases posted by certain individuals, we can determine what makes that area popular and under what conditions (e.g. recreational, academic, commercial, food related, etc.).

3.1.2 Identification/Selection

Tweets are represented as 3D objects in the environment. The status of a tweet can be represented visually by the model observed by the player. Characteristics of tweet models such as type, size, color and motion allow the player to instantly know the nature of the tweet. By default, tweets are represented as blue birds synonymous of the Twitter logo. However, if the user wants to match a tweet to a pre-processed analytic, they can customize which model and color that tweet should now look like. As per the example shown in Figure B-1, the red model of a skull depicts the matching of the word ’danger’.

These visual queues now give the player an enhanced situational awareness as they are immediately represented in the scene. Users have the option to perform further actions to further dive deeper into the dataset. During gameplay, the position of a tweet relative to the player adjusts the game object’s Level Of Detail (LOD). A spherical interaction zone highlights tweets that are in the vicinity with a radius of approximately 20 meters. When a particular tweet is selected, a 3D display is rendered showing all the original data as it was read in such as username, follower count, timestamp, text, etc. As opposed to the DK1 version where this speech bubble is positioned next to the tweet in world space (as seen in Figure 2-5), the DK2 version embeds this information display as a part of the cockpit user interface (as seen in Figure 2-9). Now a scrollable screen is within reach of the player and text elements are more easily viewed from the player.

3.1.3 Filtering/Dynamic Queries

As shown in Figure 2-5, menu options on the GUI allows for further analysis on the data. Being able to apply filters and dynamic queries can help analysts focus on specific features, reveal underlying structure, and formulate hypotheses. In this project, there are a few ways in which we can filter the Twitter data in the original domain. First, analysts can select a time range interval which narrows down the tweets in the dataset by their corresponding timestamp. Next, a user can define which models are visible, deciding whether to see all the tweets at once or just a subset. Another option is to change the physical landscape itself by adjusting the opacity of the buildings rendered in the scene. By default, buildings are fully opaque with a solid color. However, there are options to change shaders applied to the 3D model such that it is wire-framed or completely transparent. UV mapping functionality allows the model to apply Google Maps textures and make the 3D scene more realistic. Changing the shader of the campus model allows the option to compare tweets in separation or in conjunction with their landscape.

Progress has been made to conduct queries that bring tweets from a physical to a logical representation. Tweets can be searched by keywords that can produce groupings in three-dimensional space. By use of a virtual keyboard, users can type and define a criteria to do a string match on the tweets. Determining how the virtual keyboard interacts and responds to player input went through many iterations due to the margin of error that resulted from selecting small buttons. When attempting to select a specific key or character, users found it tedious trying to do a selection in a small area. The solution was to create a Top-Down Open Palm experience that followed an Approach-Proximity-Selection pattern as seen in Figure 3-2. The Leap Motion has the best hand recognition and tracking when the hand is outward facing, palm is perpendicular to the line of sight, and the fingers are spread out wide, nonoccluding one another. As the user maintains this pose, the user's fingers can act as individual cursors for more precise selections. The proximity of the hand and fingers determines which buttons light up on the keyboard accordingly. If the duration of a finger is about two seconds, the key is selected. After typing a query and if a match exists, the tweet moves from its original location to a new one where a virtual wall is formed as shown in Figure B-1. This allows analysts to see connections and relationships between various Twitter topics, locations, users, etc.

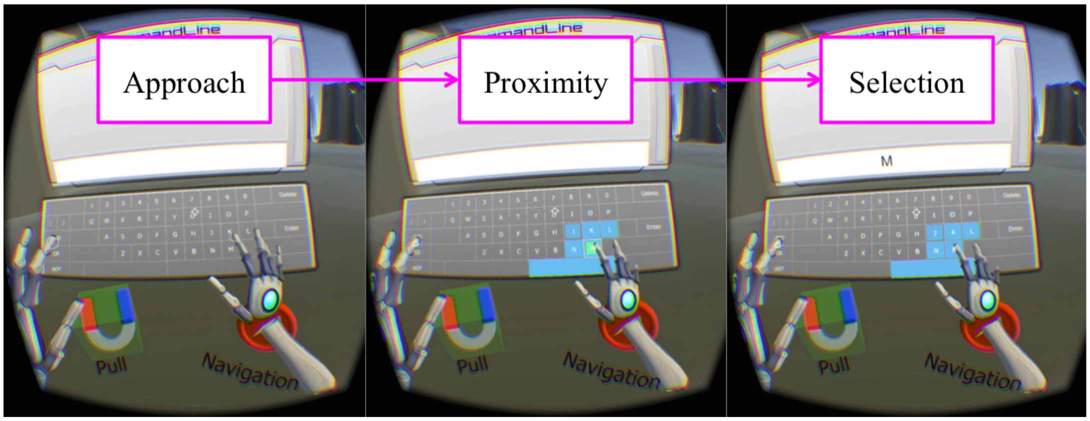


Figure 3-2: Leap Motion Interactable Keyboard
To aid a player in completing dynamic queries and filter on the Twitter dataset, a virtual keyboard was created. This followed an Approach-Proximity-Selection pattern. On Approach, the player moves there hand to desired keys on the virtual keyboard. At Proximity, specific keys light up letting the user know they are interactable. On Selection, the closest key is selected if the user's hand remains stationary long enough on that element. This then executes a key or button press.

3.1.4 Clustering/Pattern Recognition

Overlaying data on top of it's original geographical landscape can help detect patterns. For example, some tweets in this dataset share common characteristics such as location, topic, etc. In the default physical view, if a user posts a tweet at the same location of another one, the new tweet is physically placed on top of the previous tweet. As a result, vertical stacks can be created in the environment where the ordering of tweets is shown chronologically by timestamp from bottom to top. This clustering can help define the nature of the geography or the social behaviour of users. For example, clusters can be seen more around popular public places such as dining halls and dormitories on MIT's campus and less near the academic side of campus. Another noticeable pattern is that some individual users post in bursts, in which they make multiple tweets from the same location.

We can also detect patterns based on pre-processed analytics. One example of an analytic performed on the Twitter dataset was Sentiment Analysis as described in Section 2.2.2.2. By utilizing methods shown in [42] and [63], we are able to create a running count of words that is matched in a sentiment dictionary. Each word in this dictionary has a score relating to it's overall sentiment (typically ranging from -10 to 10 where the most negative\positive corresponds to the most bad\good sentiment respectively). Figure 3-3 shows an example of the scene where colors of the tweets represent their corresponding sentiment. As many are neutral in yellow, there are some noticeable areas of red (bad) and green (good). This can refer to the changing sentiment of a particular user over time (e.g. dissatisfaction over time) or of a collection of tweets at a particular location (e.g a protest or promotions on campus.). It is up to the analysts discretion to determine what context the sentiment provides.

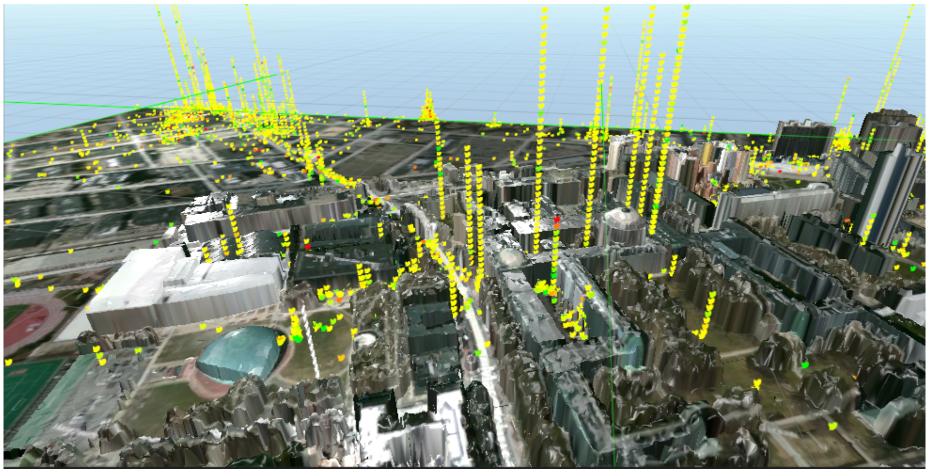


Figure 3-3: Twitter Sentiment on MIT's Campus
The Twitter dataset can be visually represented according to their sentiment score. After applying the pre-processed analytic, the color of the 3D data models can be based on a gradient from red-yellow-green corresponding to a bad-neutral-good overall sentiment.

3.1.5 Detail-On-Demand

With a tweet of interest selected, additional actions can be performed to reveal new information particular to that tweet. Hovering over and selecting the tweet with a virtual cursor opens a display in 3D space. As shown in Figure B-2, we can see all the attributes that are associated to that tweet when it was initially read into the database. Other popup windows as shown in Figure 2-9 can be utilized to extract more information pertaining to a user's post. Links mentioned within the original post can be rendered as live HTML webpages. The user's profile page can be viewed when the tweet has been selected as well. This can give more information pertaining to a specific user. Additionally, a live web browser can be rendered within the game to be used as a supplemental tool to discover more information about the user.

There are other actions that can be performed to help track user behaviour. One option is to show a user's preceding or succeeding tweet if there exists one in the dataset. Figure B-3 illustrates an example of visual pointer from one tweet to its succeeding tweet. This renders a directed 3D waypoint arrow in the scene revealing the user's next location at which they made a tweet, relative to their previous post. This helps show routes of users and known behaviours given geographical information. We can filter amongst tweets of the same user by username or by different users by hashtag.

3.2 Constructing Narrative

Executing the above analytical tasks can consequently produce the construction of a narrative. Building these stories provide a good framework for analysts to develop and test their hypotheses. With much data to digest, it can be difficult to effectively draw conclusions and develop a more direct approach to understand a network's overall structure. However, by making the execution of these analytical tasks analogous to building a story, users can create informal guidelines or objectives that can aid in the process data analysis. For example, as the player follows one tweet, it can mention a topic or a Point of Interest. Navigating to the succeeding tweet or location, the user can utilize this additional information and formulate it's reasoning based on previous posts and other factors (e.g user reputation, geographical location, etc.). These chain of events can be influenced not only by the existing flow of data but also by the analyst's natural train of thought.

These stories make the results for data analysis more user-friendly, persuasive, and more conducive to decision-making[23]. Stories can help construct a set of hypothesis that analysts could use to investigate data and enable more rigorous data analysis. The narrative is not only dictated by what they discover but how they reached that decision. These enrich the stories and lead to deeper insights. The idea is for the analysts to navigate back and forth between the data and the developing story to ensure a good balance between creative narrative and revealing analytics.

Chapter 4

Results

4.1 Evaluation

This thesis was conducted as an experimental-based research project with an ethnographical study on data analysts. For a general data scientist, analytical tasks are performed to discover underlying data structure and make progress in decision making. Given this specific context of juxtaposing social media on a familiar landscape such as the MIT academic campus, this analytical tool was able to relate to a broader audience. In particular, demographics were of faculty, staff, and predominantly the student body. Much evaluation came from tests exercised from participants in the laboratory as well as those attending minor demos.

4.1.1 VR as an Effective Workspace

Combining the design principles of a 3D user interface with the affordances provided in the user experience posed the question, “How effective is using virtual reality as a workspace for task management?". Simulating a virtual reality workbench exploits the use of 3D space that can be further tailored towards user needs. A more ergonomically designed interface matches human natural movements and perceptions that make it quicker and easier to complete distinct tasks. As a result, the spatial arrangement and grouping of interactable objects surrounding the user is pertinent for inspection and in-depth analysis.

Capturing user actions that are translated to analytical tasks requires an effective UI. By avoiding occlusion all together or creating a series of cascading windows that display textual information, a degree of categorization was added. This made recognition and recall of screen placement easier for the user. Positioning objects “with-in arms reach” made interaction more welcoming to the player. As objects were orderly placed accordingly, the use of visual queues enhanced user interaction. Simulating depth with lights, shadows, and interchangeable colors conveyed different states of the UI (e.g. active, enabled, disabled, etc.). The design of other affordances can be referred to Section 2.2.3.

Input management plays a key role in constructing a workbench conducive to task management. Accepting all possible inputs at all times helps minimize the margin of error for a direct task and expands user freedom. For example, when selecting a dock widget, a user can rely on the look direction of the Rift where a raycast will determine if an object is in the line of sight. In addition, the user can use the gamepad controller to confirm the selection of a widget. Using the leap motion controller, the player can alternatively use their hand to simulate touching the 3D widget.

However, some drawbacks were evident when users were trying to use this workspace. Tasks took longer to complete due to the subtlety of the interface. To accurately portray a selection, techniques recognizing hand placement and stationary movement was necessary. By maintaining the "Open Palm - Outward Facing" approach, the hand is best recognized as a 3D cursor where the fingertip edges or the center of the palm resembled a 3D cursor. Also, some tasks were not as customizable. Incorporating more tactile selections required larger text and larger items. Therefore, each pane or window had limited information and more navigation was required to go through the windows.

4.1.2 VR as a Visual Analytics Tool

Evaluating the virtual reality platform as a visual analytics tool requires close observation and knowledge of the data analysis process. Users are instructed to complete analytical tasks that assist in the exploration and understanding of the Twitter dataset. As described in Section 3.1, interaction techniques that users can complete for analysis are subject but not limited to navigation, identification, filtering, and clustering. As these tasks are executed, we could examine and make observations concerning the user's incentive. Informal performance measures can determine how effective VR is as a visual analytics tool. This includes time it takes to recognize an anomaly in the dataset (e.g. red skull model), execute a command, transition between tasks, and focus on particular aspects of the user interface. We can also take note on the scope and consistency tasks were performed, especially for navigation. All together, these tasks define activity patterns that are used to characterize each session of a user. We can then compare how consistent these metrics are amungst users over time.

Utilizing the virtual reality platform provides many advantages for effective gameplay and analysis. The 3D environment is very appealing to a first person perspective due to the extent of adaptable view modes. From a first person POV, a user is free to navigate the scene from the ground level. Transitioning to a bird's eye view, the player can see clusters and overall patterns on campus at a much larger scale. Also, it allows for supporting other strategies to discover particular insights on the dataset. In a graphical sense, VR enables the player to navigate a scene and rely on visual cues such as clustering, color, model, etc. to grab their attention and draw connections. The user can confirm relationships among user behaviour, geographical location, or temperament of conversation. Users can use this new information to guide themselves on what tasks to perform next. Furthermore, constructing the workspace to be embedded within the virtual environment is conducive for decision making. The analysts do not have to disengage themselves to complete further investigation on another application or device. Use of the in-game web browser allows for external lookups without leaving the game. Lastly, the workbench is organized to be more welcoming to natural interaction and cater to user tasks. This includes exploration, filtering, querying, and navigation. Having these readily available at the users' fingertips promotes a series of actions that are more fluid during gameplay. By limiting the number and position of displays viewed by users, they are able to be more efficient and focus on one task at a time.

There were some disadvantages using virtual reality as a data tool. When using the visualization, discovering elements that were far from the camera were difficult to see. Incorporating beacons or light-ups could help mitigate this issue. Navigation also remains a concern for users during gameplay. Teleporting to cached destinations is a simple solution, however, does not facilitate more user freedom to navigate to other regions of interest in the scene. Also, there are a limited number of actions that can be performed at a given time. Users would have to take additional steps to accomplish a series of tasks. Lastly, there are only a few tasks available to the player total. Given that this was an experimental tool, only a few options were implemented initially. Additional tasks would need to be made available for users to more accurately depict the data analysis process.

4.2 Performance

4.2.1 Ingest on Database

Performance and high frame rate is important when working in simulations that show many data points. Ingesting the Twitter data on Accumulo with D4M analytics is proven to be fast. D4M achieved 100,000,000 inserts per second as it's peak performance[42]. Accumulo leverages HadoopDFS[8], which is an open source replicated block based distributed filesystem modeled after Google Big Table[25]. This database is typically NoSQL, allowing for the construction of data mining applications that do little read-modify-write and contain relaxed restrictions for performance boosts.

```
% use pMatlab to define global indices and map the files
myFiles = global_ind(zeros(Nfile,1,map([Np 1],'c',0:Np-1)));
```

Figure 4-1: Command to Execute pMatlab Function in Parallel
pMatlab is used to define global indices and map the files defined by Nfile. Np is used to define the specific number of processors to use.

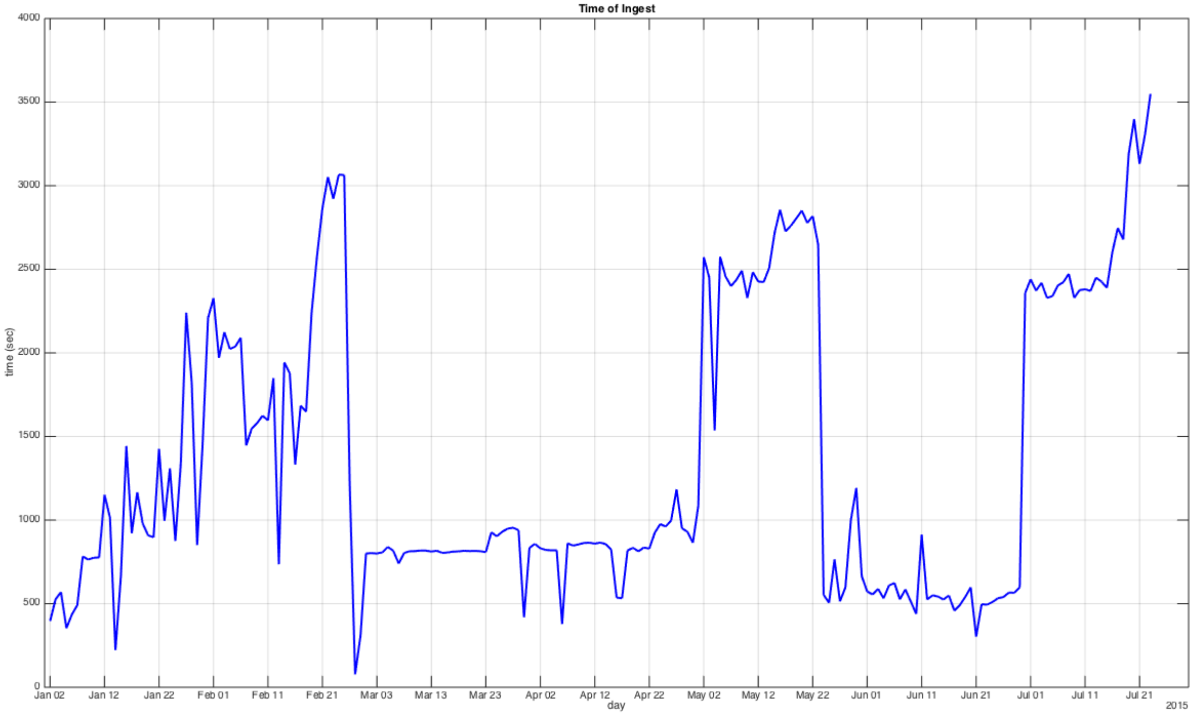


Figure 4-2: Time of Ingest - Daily
Ingestion of Twitter data that extracted tweets in the range of MIT. Tweets were collected daily from January 1, 2015 to July 25, 2015. Months of higher data ingest include February, May, and July.

Most of the computation comes from parsing the pre-processed Twitter data. We utilized a mechanism called distributed arrays that are useful for writing efficient parallel programs[41]. By defining a specific number referring to the processor the instance is running on, we can run algorithms in parallel. In particular, we added a “map" object to the construction of an array in pMatlab. A single command demonstrating the mapping is shown in Figure 4-1. Figure 4-2 shows the overall time

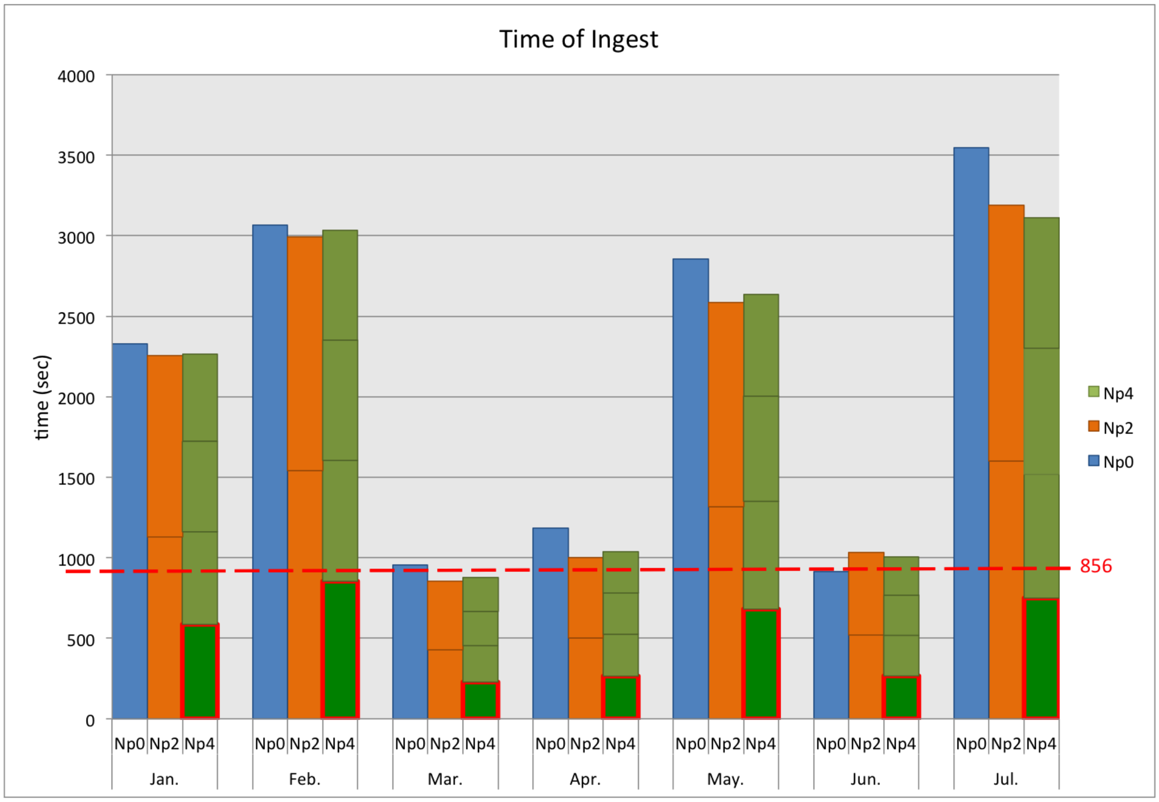


Figure 4-3: Time of Ingest - Parallel
Comparison of times when ingesting the data with different number of processors, $N_p = 0$(unparalleled - blue), 2(orange), and 4(green). When running with four processors, the maximum time was about 856 seconds, approximately 14 minutes.

when ingesting the data for the course of seven months from January 1, 2015 to July 25, 2015. Figure 4-3 reflects the total time for each job to execute when running under a specified number of processors $N_p = 0$(unparalleled), 2,and4. As expected, running separate jobs with an increased number of processors linearly decreases the time in which the job finishes.

4.2.2 Game Rendering and LOD

Creating a 3D environment can be costly on both the GPU and CPU. As the camera renders a scene, the number of draw calls is determined by the amount of faces and vertices that are within the field of view. Therefore, the higher resolution the 3D model, the more workload is performed on the processor. When instantiating 3D objects with colliders, additional computation is needed for placement and collision

Table 4.1: Level of Detail Performance (FPS)

	1000	5000	10000	25000	50000	100000	250000	500000
Particles	745	411	304	138	92	49	18	6
Cull 0%	22	3	1	-	-	-	-	-
Cull 1%	157	32	21	8	3	-	-	-
Cull 2%	250	68	37	14	5	2	-	-

Recorded rate in frames per second (FPS) based on object instantiation type and number of objects viewable from the player's camera. Particles do not contain any 3D object geometry. Therefore, more workload is done on the GPU and there is a more noticeable performance improvement. Culling is used to render 3D objects when only a percentage of the object is within the camera's FOV. As expected, the higher the percentage, the better the FPS and more objects can be viewed.

detection. Billboarding, which attaches a script that does auto-rotation towards the camera, can also add a performance hit during runtime.

One solution in Unity3D practice to improve performance was to take advantage of particle systems and Level Of Detail (LOD)[71]. Particle systems are techniques in computer graphics and game physics that uses a large number of sprites or meshes to render and resemble a collective entity. The shader that is drawn on the GPU defines features of each particle. For example, matrix calculations can reference the particle's pose relative to the global camera in order to reposition the particle to face the camera. However, there are limitations on binding data to locations where particles are rendered. LOD allows the instantiation and resolution of a 3D object with a user-specified location to occur relative to the player's distance from that object. If the object is outside the vicinity of a fixed distance from a player, the object is culled and no longer rendered from the camera. Hence, user interaction is limited within the player's field of view and object states such as collider activation can be more easily controlled[33]. Section 4.2.2 compares the total number of objects viewed with the type of object instantiation. Culling percentage is determined by the proportion the object is seen on the screen.

Chapter 5

Conclusion

5.1 Summary

One of the main challenges with Big Data today is coming up with a proper data representation for efficient user analysis. As data scales into higher dimensions, it can become overly complex. Visualization is key in the improvement of pattern recognition and data analytics. At Lincoln, we experimented with using novel methods and emerging technologies to enhance visualization and user interaction for data analysis. Virtual reality creates an immersive environment for the user; as data is overlaid within a geographical domain, an enhanced situational awareness and cognition can be achieved.

These advances in virtual reality continues to grow as computation and processing becomes faster on both the hardware and software fronts. As a result, these devices are becoming more powerful, affordable, and readily available to the research and development community. This increases the capability of integrating visual data exploration and interaction within VR. Below, we address some concerns during the development process and potential avenues moving forward.

5.2 Challenges and Areas of Improvement

5.2.1 Hardware

For demo and portability interests, this work has been completed on a Macbook Pro laptop. Although producing promising results, there were some foreseeable limitations. As more objects populate the scene, more system checks are completed frame by frame. It is recommended to have a faster processor to achieve better performance and reduce jerky movement as the camera pans a scene (e.g. scene judder). Oculus suggests a frame rate of at least 60-75 fps for a comfortable user experience. With more vertices rendered in the scene, more draw calls are sent to the GPU. In addition, Oculus Rift rendering and Leap Motion gesture recognition requires a lot of processing.

In May 2015, Oculus development for the OS X and Linux paused in order to focus on delivering the high quality consumer-level VR experience for Windows[35]. The Rift requires a desktop-level graphics processor. The 15-inch Macbook Pro uses mobile graphic processors that don't necessarily have the processing power of the desktop graphics cards in the Rift's preferred specs. Upgrading from a traditional laptop to a more powerful machine can produce a higher frame rate and a more ideal game experience.

5.2.2 Usability

"Know thy user, for he is not thee"

Conducting experimental research in which the lead developer makes core decisions can misinterpret what the user wants and how the product is designed[58]. Partially falling into the same demographic of a data scientist, I have based my decisions towards my experience or qualitative results proven in practice that seemed the most suitable. Having bias that detracts from having complete merit towards the user can impact the full potential of a user product. However, given that this was an experimental study, this did not hamper my development process. Since most of the project was defining an effective user interface as opposed to iterating on an improved product, the research enabled me to develop more freely. In addition, most participants and active users were not very experienced or professional analysts. A more involved study including experts in data analysis could have proven to be very useful in evaluating the effectiveness of this platform.

Designing a 3D user interface in virtual reality remains an experimental and investigative study. Instructing users to break traditional 2D conventions when utilizing computer interfaces and adapting to a 3D virtual display that appeals more to human ergonomics is a strong transition. The creation of a virtual workbench was essential to the visual analytics pipeline. The goal was to create a task management system that continues to guide the user, supports different analytical strategies, caters to user demands, and facilitates further exploration. As described in Section 2.2.3, this requires an effective user interface that makes objects and items more interactable and responsive. It was necessary to create supplemental visual queues to simulate both depth and positioning. For example, the arrangement and labeling of analytical tasks as 3D widgets seemed to be an improvement on the 2D task bar that usually remained fixed to the bottom of the user's camera.

Now that we are creating a sense of telepresence, it's essential to not break immersion. However, some constraints ephasized some drawbacks that effected usability. One disadvantage includes the processing power of CPU and GPU. When there is a noticeable latency in frame rate, it's hard to maintain fluid gameplay and scene judder is more prominent. Another is potential delay in input to output responses, such as selecting items and monitoring head movement. Navigation also remains an issue due to limited input devices that can be convincingly portrayed in the virtual environment. In addition, there is a limited number of user actions, which confines user freedom. Making more iterations that enhance the availability of these tools could have a considerable impact.

5.3 Future Work

Although much progress has been made, further improvements could enhance both application performance and user interactive gameplay. Rendering 3D models scales linearly with performance. Activating and deactivating colliders when needed can help reduce the computation load. Additional shaders could be applied to the 3D buildings of Cambridge to provide a better rendition and give the player more options of how the tweets are overlaid in the scene. Occlusion layers for overlapping tweets and blocked buildings could be applied to prevent unnecessary rendering.

Further optimization can be performed on the campus model. The raw LADAR data produced a model of about 200,000 vertices for each of the 15 sections, totaling roughly three million vertices. We could construct a point cloud mesh where each vertex is drawn directly on the GPU. This allows an order of magnitude of millions of points to be generated without maintaining the overhead of rendering additional faces and other geometry. However, this is determined by the shader used and there is no intractability of the 3D positional data during runtime. For a gameplay experience, we reduced the vertice count nearly 30 percent to 100,000 vertices. Then, the formation of faces allowed for a common static environment to be constructed with features such as colliders and occlusion culling.

Additional optimization can be performed by displaying a dynamic environment and how to improve the mapping of tweets. To make the 3D setting more scalable, we could incorporate dynamic terrain construction that incorporates Shuttle Radar Topography Mission (SRTM) data and Google 3D Maps. To be more accurate, we could use a Mercator projection that does the same transformation of latitude and longitude coordinates as the 2D Google Maps textures. Also, we could further utilize the geohash to group tweets more precisely in accordance to the distance of the player. Therefore, the resolution of viewable tweets can be displayed in a grid layout.

Although we have a few useful analytics now, we intend to add more features that allow for further engagement by the player. Originally, this work was done in Unity 4.6 and Oculus Rift DK1. Implementing Unity's UI system allows for 3D text and more engagement with Leap Motion. Continuing to exercise 3D interactions from hand inputs rather than gamepad controllers could help immerse the player and manipulate the data more effectively. Currently, we have upgraded to Unity 5.0 and Oculus Rift DK2 to utilize the enhanced display and more accurate positional head tracking. Some potential future features we plan to implement in the user-interface include multi-selection and annotation. Algorithms using machine learning and sentiment analysis techniques could be used to further analyze the data. We also plan to continue researching other ways to enable user interaction and improve usability.

5.4 Closing Remarks

This project reveals the added potential of how utilizing the VR platform can bring a more effective visual experience. We have effectively visualized Twitter on a 3D model of MIT's campus to improve Big Data visual analytics. This research has shown how

1. Virtual reality can also be used as a data visualization platform
2. Creating a more immersive environment enables user interaction
3. Patterns and visual analytics are more efficient when working in a geospatial domain.
4. Design choices promote the improvement of visual analytical systems

As virtual reality and other technologies continue to improve, these mediums are highly considered in the pursuit for effective data visualization and enhanced situational awareness. With this research of showing tweets on MIT's campus, efforts can be made to extend this work into using other related geo-tagged information that can be embedded into an interactive 3D world.

Appendix A

Equipment Specifications

Image	Hardware	Software
	• MacBook Pro 15-inch, Mid 2009 • Processor: 3.06 GHz Intel Core 2 Duo • Memory: 8 GB 1067 MHz DDR3 • Graphics: NVIDIA GeForce 9600M GT 512 MB	• OSX 10.10.5 Yosemite • Unity3D 5.1.1f1 • MonoDevelop 4.0.1
	• Oculus Rift DK2	• SDK 0.5.1 Beta
	• LeapMotion Controller	• SDK 2.3.1+31549
	• XBOX 360 Controller (wired)	• TattieBogle XBOX 360 Driver[18]

Table A.1: Equipment Hardware and Software Specifications

Appendix B

Additional Screenshots

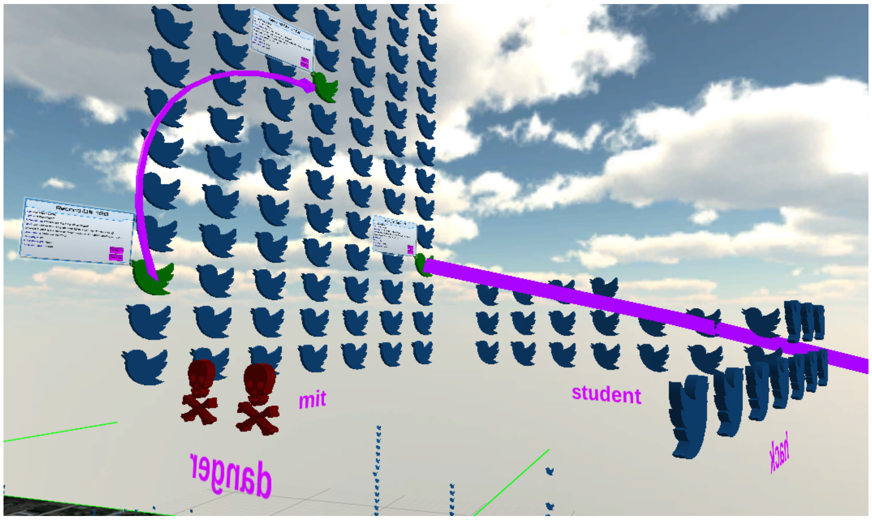
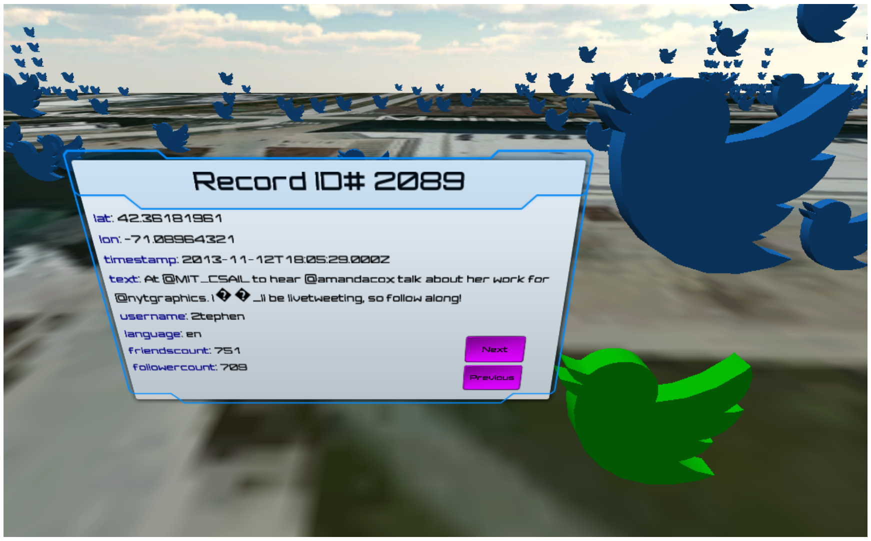


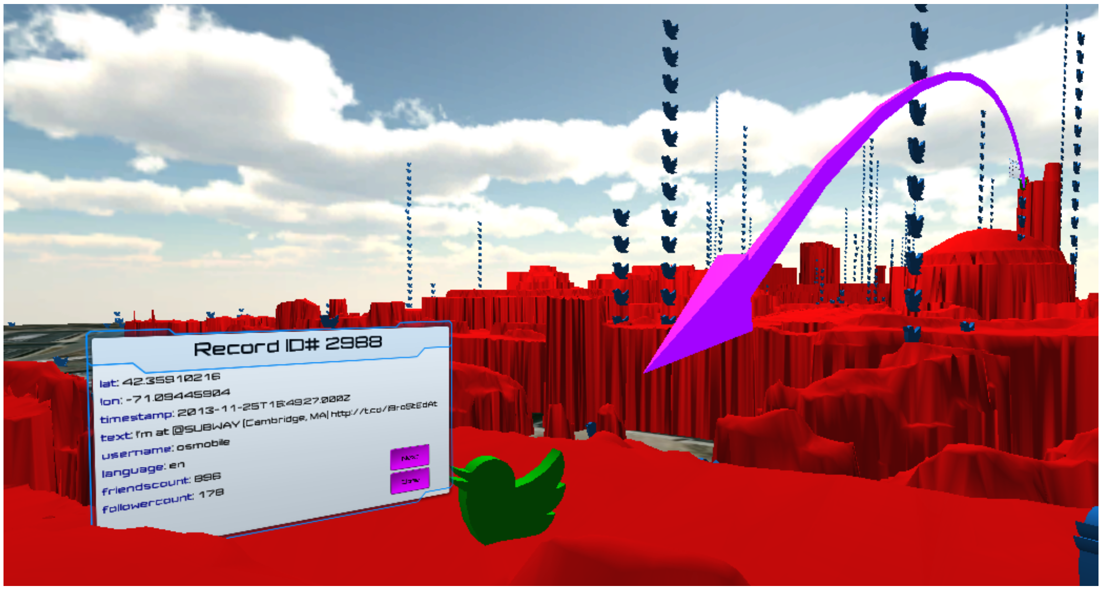
Figure B-1: Screenshot of Multiple Queries
Queries can be performed on the dataset to create a floating virtual room where walls are populated by tweets that match user defined criteria.



Record ID# 2089

lat: 42.36181961
lon: -71.08964321
timestamp: 2013-11-12T18:05:29.000Z
text: At @MIT_CSAIL to hear @amandacox talk about her work for @nytgraphics. I� �_ll be livetweeting, so follow along!
username: 2tephen
language: en
friendscount: 751
followercount: 709

Figure B-2: DK1 Information Overlay on Tweet
Upon selection, the 3D representation of a tweet changes color and launches a speech bubble revealing all characteristics.



Record ID# 2988

lat: 42.35910216
lon: -71.09445904
timestamp: 2013-11-25T16:49:27.000Z
text: I'm at @SUBWAY (Cambridge, MA) http://t.co/BroStEdAt
username: osmobile
language: en
friendscount: 886
followercount: 178

Figure B-3: Waypoints for Tweet Posts
Waypoint arrows rendered in the virtual world lets the player track social behaviour of Twitter users in the order in which the tweet was delivered.

Appendix C

Associative Arrays and D4M[1]

C.1 Associative Arrays

Associative arrays are data structures that can perform mathematical operations and represent complex datasets. They can be used to show the associations between multidimensional entities (e.g. row, column, and value tuples). Some key features of associative arrays are as follows:

- From d sets of keys $K_1 \times K_2 \times ... \times K_d$, they map to a value set V.
- They are similar in structure as matrices; consist of row and column keys as strings and values represented as strings or numbers.
- As a data structure, associative arrays return a value given some specified number of keys.

  e.g, $A(K_1) = v_1$, where A is an Associative Array.
- They are defined as algebraic semi-rings.

  A ring R is a set with two main operations: (1) Addition (which maintains the mathematical properties of associativity, commutativity, additive identity, and additive inverse) and (2) Multiplication (which has the properties of associativity and the multiplicative identity).

[1] All notes are adapted from MITLL section course on Advanced Database Technologies[56]

Therefore, a semi-ring is a set R with all the properties of a ring except an additive inverse. Also, the distributive property is valid for rings and semi-rings.

- Closed under algebraic and set operations

  e.g, `A+B`, `A−B`, `A&B`, `A|B`, `A⋆B`, etc. all yield an Associative Array.

- Array indexing is composable.

  e.g, `A(1:2,:)`, `A == '1'`, etc. update or reproduce an Associative Array.

C.2 Two-Dimensional Associative Arrays

In the 2D case, two keys (known as the row and column) can map to one value. Figure C-1 shows an example of the two-dimensional associative array. These arrays can have multiple representations for complex datasets such as the following:

- A sparse matrix with string row and column labels.

- A graph with vertex and edge labels or weights.

- 1-to-1 triple store, as utilized in the Accumulo database.

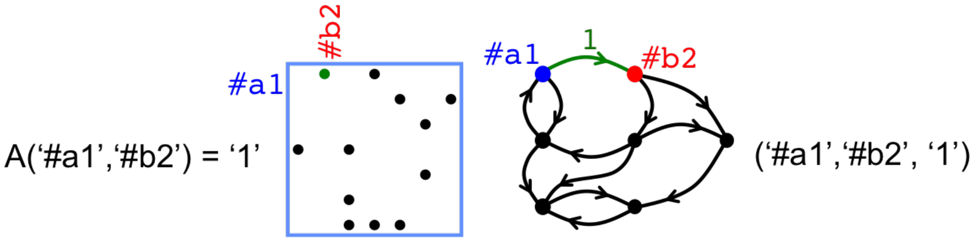


Figure C-1: Two-Dimensional Associative Arrays Example

In this example, Associative Array A has two keys called row and column keys. The value for row key $a1$ and column key $b2$ is the string 1. Associative arrays can be thought of as sparse matrices where the value maps to the intersection of the row and column labels.

C.3 D4M

C.3.1 Introduction

Dynamic Distributed Dimensional Model (D4M) is a library that allows you to represent data as Associative Arrays. These Associative Arrays are manipulated using standard linear algebraic operations. Databases like Accumulo can utilize Associative Arrays and can be implemented in MATLAB or Octave. The D4M API makes it very easy to develop analytics, perform calculations, and undergoe indexing on Associative Arrays.

C.3.2 Indexing and Querying

Works like the sparse matrix data structure in MATLAB, but with string (character array) keys. After indexing and querying, the result always returns an associative array. Below are some simple commands and operations that can be executed:

- Index into the Associative Array using row and/or column key

  e.g, `A('#al,', '#b2,')`

- Every label ends with a delimiter (e.g `','`) to allow the concatenation of keys

  e.g, `A('#al,:#d4,', '#b2,')`

- Use “:" to extract all the elements or a specified range

  e.g, `A('#al,:#d4,', :)`

- Use “StartsWith" to indicate all keys that start with a given substring

  e.g, `A(StartsWith('#al,'), :)`

- Can also use integers to index a subset of the Associative Array

  e.g, `A(1:5, :)`

- Use “>",“<", “>=", “<=", and “==" to get sub-Associative Array with values that satisfy the given condition e.g, `A>B`, `A≥B`, `A<B`, `A≤B`, `A == B`

C.3.3 Constructing and Destructing

Utilizing the D4M API, there are certain ways in which Associative Arrays can be constructed and destructed. Below lists some examples in which information can be obtained from an existing Associative Array or how a new one can be generated.

- Use “find" to extract associative array triples

  e.g, `[rows, columns, values] = find(A).` Results in a three character arrays containing row labels, column labels, and values. One for each separate entry, ending with a specified delimiter.

- Use “NumStr" to get the number of keys or values in any character array

  e.g, `NumStr(rows)`

- Use “Row", “Col", “Val" to get character arrays of unique row/column keys and values

  e.g, `Row(A)`, `Col(A)`, `Val(A)`

- Construct an associative array using “Assoc"

  e.g, `A = Assoc(rows,columns,values).` Input parameters are strings of row/column keys and their respective values. Each input parameter should have one or more of the same number of keys and values.

C.3.4 Communication with Database

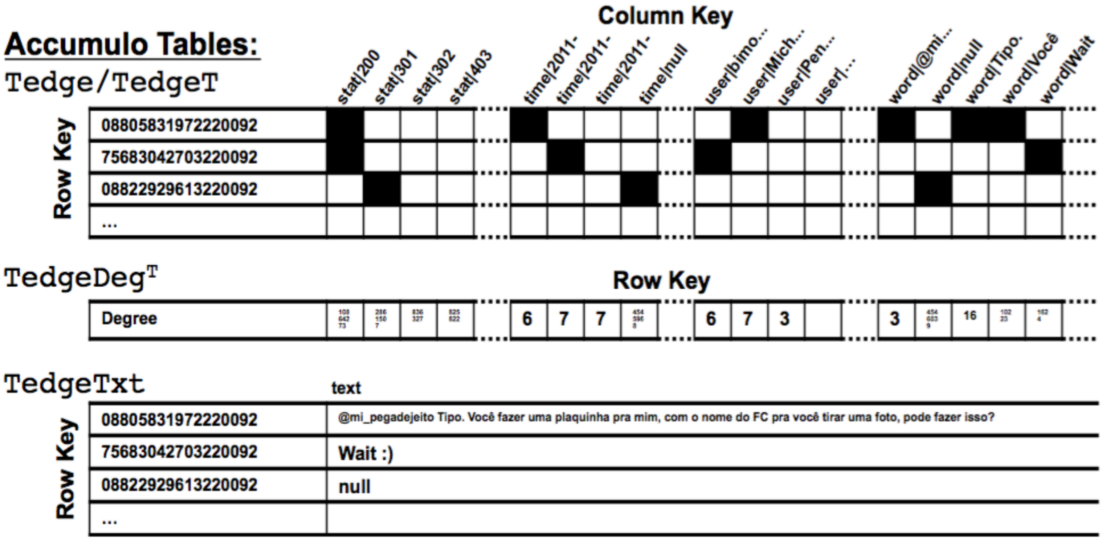
It is important to know how to extract keys and triples of an associative array, and how to construct them.

- Connect to database using “DBServer" and host credentials

```
DB=DBserver(hostName,'Accumulo',instanceName)
```

- Bind data to Accumulo tables

```
Tedge=DB('tweets_Tedge', 'tweets_TedgeT');
TedgeTxt=DB('tweets_TedgeTxt');
TedgeDeg=DB('tweets_TedgeDeg');
```



Row Key	stat|200	stat|301	stat|302	stat|403	time|2011-	time|2011-	time|2011-	time|null	user|bimo...	user|Mich...	user|Pen...	user|...	word|@mi...	word|null	word|Tipo.	word|Você	word|Wait
08805831972220092	■				■					■			■		■	■	
75683042703220092	■					■			■								■
08822929613220092		■						■						■			
...																	

Degree	108 642 73	286 150 7	836 327	825 822	6	7	7	454 596 8	6	7	3		3	454 603 9	16	102 23	162 4

	text
08805831972220092	@mi_pegadejeito Tipo. Você fazer uma plaquinha pra mim, com o nome do FC pra você tirar uma foto, pode fazer isso?
75683042703220092	Wait :)
08822929613220092	null
...	

Figure C-2: Representation of Accumulo Tables
Representation of tables as used on the Accumulo database: (1)Tedge is a table that is labeled by row keys corresponding to tweet ID and column keys corresponding to corresponding tweet data. If a relationship exists between a row and column key, their value is represented as a boolean, (2) TedgeT is the transpose of table Tedge, (3) TedgeDeg is the table that corresponds to the sum of unique column/value pairs, and (4) TedgeTxt contains the original tweet text.

- Use the “nnz" command to return number of entries in the table.

  e.g, `nnz(Tedge)`

- Use “put" to insert Associative Array in a table

  e.g, `put(Tedge,A)`

- Querying table is the same as indexing into an Associative Array

```
Atmp=Tedge(:,'word|#a1');
A=Tedge(Row(Atmp),:);
```

C.3.5 Sentiment Example

Sentiment analysis for tweets can be completed by representing Associative Arrays as sparse matrices. Below, we illustrate this example with the following steps:

1. Create a smaller sentiment Associative Array (S) containing a single column "score" for each row representing a single word. This is constructed from a sentiment dictionary with a weight corresponding to each word (e.g. `'AFFIN-111.txt'` is a viewable online sentiment dictionary used for analysis[63]).

2. Extract all the columns in the original Twitter Associative Array that contain any of the words in the sentiment Associative Array, forming a new Associative Array (W).

3. Do a summation by multiplying the logical representation of the W with the values of S to construct a new Associative Array that contains a score for each tweet.

The result can simply be appended to the original Associative Array, augmenting it with a new sentiment column. The function call in MATLAB is shown in Figure C-3 and the visual representation of the Associative Array matrix manipulation and arithmetic can be seen in Figure C-4.

```
% DetermineScore - Logic for Sentiment Associative Array
%% Returns final associative array, Aout, which has the appended
%% sentiment score determined from the initial associative array, A
function [ Aout ] = DetermineScore( A )
    %Construct assoc from sentiment dictionary
    S = CreateSentimentAssoc('AFFIN-111.txt', 'word_lower', 'score');
    W = A(:,Row(S));                    %Match sentiment words
    Anew = dblLogi(W)*str2num(S);       %Summation of scores
    Aout = A + Anew                     %Update final assoc
end
```

Figure C-3: Determine Sentiment Score for Twitter Data in Associative Array Function call in MATLAB that appends a sentiment score column to an Associative Array containing Twitter data.

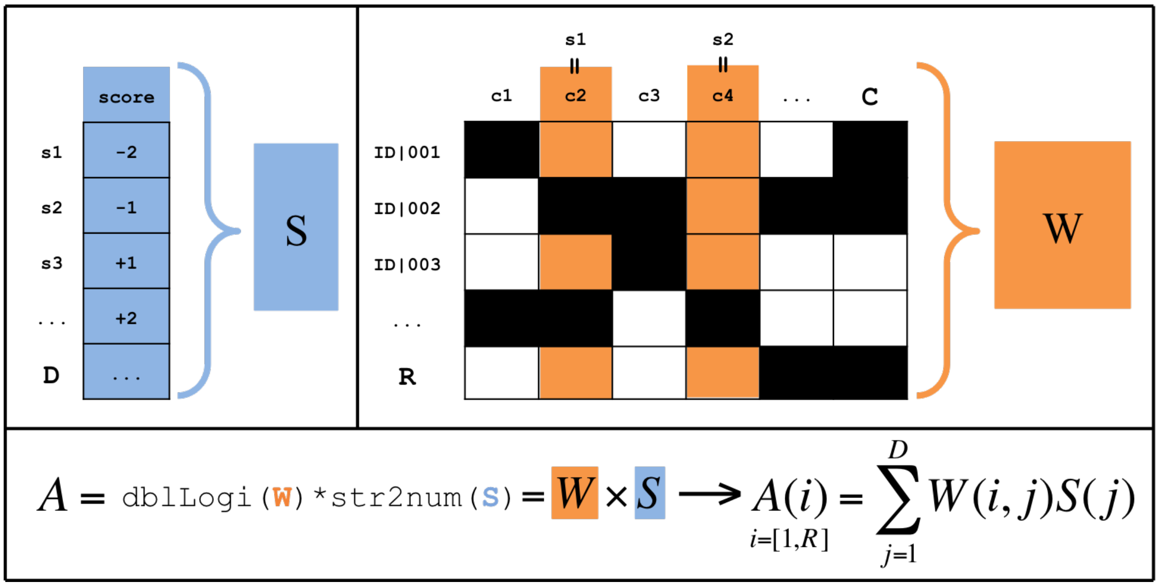


Figure C-4: Matrix Representation of Sentiment Example

Sentiment analysis for Twitter example can be viewed in matrix form as the formation of a sentiment dictionary Associative Array (S) and the subset of corresponding tweets that match to the words found in the sentiment dictionary (W). The formation of the new Associative Array (A) with a sentiment score is a simple matrix multiplication between W and S.

Bibliography

[1] 3D Warehouse. https://3dwarehouse.sketchup.com/.

[2] Apache Accumulo. https://accumulo.apache.org/.

[3] Blender. http://www.blender.org/.

[4] D4M: Dynamic Distributed Dimensional Data Model. http://www.mit.edu/~kepner/D4M/.

[5] GNIP. http://gnip.com/sources/twitter/.

[6] GNIP Historical API. http://support.gnip.com/apis/historical_api/overview.html.

[7] Google Earth. http://www.google.com/earth/.

[8] Hadoop Distributed File System. http://www.aosabook.org/en/hdfs.html.

[9] Leap Motion. https://www.leapmotion.com/.

[10] MAPD. mapd.csail.mit.edu.

[11] Maya. http://www.autodesk.com/products/maya/overview.

[12] Microsoft Kinect. https://dev.windows.com/en-us/kinect.

[13] MIT Computer Science and Artificial Intelligence Laboratory. https://www.csail.mit.edu/.

[14] Oculus Documentation: Best Practices. https://developer.oculus.com/documentation/intro-vr/latest/concepts/bp_intro/.

[15] Oculus Rift. https://www.oculus.com/.

[16] pMatlab. http://www.ll.mit.edu/pMatlab/.

[17] Samsung Gear VR. http://http://www.samsung.com/us/explore/gear-vr/.

[18] TattieBogle XBOX 360 Driver. https://github.com/360Controller/360Controller.

[19] Unity3D. http://unity3d.com/.

[20] Unite Boston 2015. https://unity3d.com/events/unite-2015-boston, 2015.

[21] Rony Abovitz, Brian T Schowengerdt, and Matthew D Watson. Planar waveguide apparatus with diffraction element (s) and system employing same, April 24 2015. US Patent App. 14/696,347.

[22] Mike Alger. Visual Design Methods for Virtual Reality. http://aperturesciencellc.com/vr/VisualDesignMethodsforVR_MikeAlger.pdf, 2015.

[23] Judy Bayer and Marie Taillard. Story-driven Data Analysis. *Harvard Business Review*, 2013.

[24] Raluca Budiu. Memory Recognition and Recall in User Interfaces. *Nielsen Norman Group*, 2014.

[25] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C Hsieh, Deborah A Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, and Robert E Gruber. Bigtable: A distributed storage system for structured data. *ACM Transactions on Computer Systems (TOCS)*, 26(2):4, 2008.

[26] Hsinchun Chen, Roger HL Chiang, and Veda C Storey. Business Intelligence and Analytics: From Big Data to Big Impact. *MIS quarterly*, 36(4):1165–1188, 2012.

[27] Jian Chen, Haipeng Cai, Alexander P Auchus, and David H Laidlaw. Effects of stereo and screen size on the legibility of three-dimensional streamtube visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2130–2139, 2012.

[28] Peter Cho, Hyrum Anderson, Robert Hatch, and Prem Ramaswami. Real-time 3D ladar imaging. In *Applied Imagery and Pattern Recognition Workshop, 2006. AIPR 2006. 35th IEEE*, pages 5–5. IEEE, 2006.

[29] Alex Chu. VR Design: Transitioning from a 2D to a 3D Design Paradigm, 2014.

[30] Ernest Cline. *Ready Player One*. Broadway Books, 2011.

[31] Carolina Cruz-Neira, Daniel J Sandin, Thomas A DeFanti, Robert V Kenyon, and John C Hart. The CAVE: audio visual experience automatic virtual environment. *Communications of the ACM*, 35(6):64–72, 1992.

[32] SG Djorgovski, Piet Hut, Rob Knop, Giuseppe Longo, Steve McMillan, Enrico Vesperini, Ciro Donalek, Matthew Graham, Asish Mahabal, Franz Sauer, et al. The MICA Experiment: Astrophysics in Virtual Worlds. *arXiv preprint arXiv:1301.6808*, 2013.

[33] DocktorAce. Remove collider vs putting on a layer that ignores collisions. answers.unity3d.com/questions/209582/remove-collider-vs-putting-on-a-layer-that-ignores.html#answer-244468, April, 25 2012.

[34] Ciro Donalek, SG Djorgovski, Alex Cioc, Anwell Wang, Jerry Zhang, Elizabeth Lawler, Stacy Yeh, Ashish Mahabal, Matthew Graham, Andrew Drake, et al. Immersive and collaborative data visualization using virtual reality platforms. In *Big Data (Big Data), 2014 IEEE International Conference on*, pages 609–614. IEEE, 2014.

[35] Jackie Dove. Oculus Rift drops Mac and Linux, targets Windows PCs only. http://thenextweb.com/dd/2015/05/15/oculus-rift-drops-mac-and-linux-targets-windows-pcs-only/, 2015, May 15.

[36] Jon Favreau. *Iron Man 2.* Marvel Studios, Fairview Entertainment, 2010.

[37] Matt R Fetterman, Zachary J Weber, Robert Freking, Alessio Volpe, D Scott, et al. LuminoCity: a 3D printed, illuminated city generated from LADAR data. In *Technologies for Practical Robot Applications (TePRA), 2014 IEEE International Conference on*, pages 1–4. IEEE, 2014.

[38] Quan Ho and Mikael Jern. Exploratory 3D Geovisual Analytics. In *Research, Innovation and Vision for the Future, 2008. RIVF 2008. IEEE International Conference on*, pages 276–283. IEEE, 2008.

[39] Matthew Hubbell and Jeremy Kepner. Large Scale Network Situational Awareness Via 3D Gaming Technology. In *High Performance Extreme Computing (HPEC), 2012 IEEE Conference on*, pages 1–5. IEEE, 2012.

[40] Daniel A Keim. Information Visualization and Visual Data Mining. *Visualization and Computer Graphics, IEEE Transactions on*, 8(1):1–8, 2002.

[41] Jeremy Kepner. *Parallel MATLAB for multicore and multinode computers*, volume 21. SIAM, 2009.

[42] Jeremy Kepner, William Arcand, William Bergeron, Nadya Bliss, Robert Bond, Chansup Byun, Gary Condon, Kenneth Gregson, Matthew Hubbell, Jonathan Kurz, et al. Dynamic distributed dimensional data model (D4M) database and computation system. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 5349–5352. IEEE, 2012.

[43] Blazej Kot, Burkhard Wuensche, John Grundy, and John Hosking. Information visualisation utilising 3D computer game engines case study: a source code comprehension tool. In *Proceedings of the 6th ACM SIGCHI New Zealand chapter's international conference on Computer-human interaction: making CHI natural*, pages 53–60. ACM, 2005.

[44] Wolfgang Krüger, Christian-A Bohn, Bernd Fröhlich, Heinrich Schüth, Wolfgang Strauss, and Gerold Wesche. The responsive workbench: A virtual work environment. *Computer*, (7):42–48, 1995.

[45] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is Twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, pages 591–600. ACM, 2010.

[46] Jaron Lanier. Virtual Reality: The Promise of the Future. *Interactive Learning International*, 8(4):275–79, 1992.

[47] Steve Lohr. The age of big data. *New York Times*, 11, 2012.

[48] James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, and Angela H Byers. Big data: The next frontier for innovation, competition, and productivity. 2011.

[49] Bernard Marr. Big Data: using SMART big data, analytics and metrics to make better decisions and improve performance, 2015.

[50] Andrew McAfee and Erik Brynjolfsson. Big Data: the management revolution. *Harvard business review*, (90):60–6, 2012.

[51] Bill Meacham. Perception and Reality. *Philosophy for Real Life*, 2015.

[52] Jody Medich. What Would a Truly 3D Operating System Look Like? `http://blog.leapmotion.com/truly-3d-operating-system-look-like/`, 2015, April 25.

[53] Austin Modine. Apple patents OS X dock. *The Register*, 2008.

[54] Gordon Moore. Moore's Law. *Electronics Magazine*, 1965.

[55] Leap Motion. VR Cockpit. `https://developer.leapmotion.com/gallery/vr-cockpit`, 2015, July 8.

[56] Julie Mullen, Lauren Edwards, and Vijay Gadepally. Advanced Database Technologies: System Challenge: D4M. In *Big Data Advanced Database Technologies. IEEE Boston Section Course*. IEEE, Spring 2015.

[57] Donald A Norman. *The design of everyday things: Revised and expanded edition.* Basic books, 2013.

[58] David S Platt. *Why Software Sucks–and what You Can Do about it.* Addison-Wesley Professional, 2007.

[59] Adi Robertson. Oculus rift virtual reality head mounted display kickstarter. *The Verge*, 2012.

[60] B Schneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Visual Languages, 1996 Conference on*, pages 336–343. IEEE, 1996.

[61] Michael Schroeck, Rebecca Shockley, Janet Smart, Dolores Romero-Morales, and Peter Tufano. Analytics: The real-world use of big data. *IBM Institute for Business ValueâĂŤexecutive report, IBM Institute for Business Value*, 2012.

[62] Orson Scott Card. *Ender's Game*. Tor Books, 1985.

[63] Loren Shure. Analyzing Twitter with MATLAB. *Matlab Central*, 2014.

[64] Steven Spielberg. *Minority Report*. Ablin Entertainment, Cruise/Wagner Productions, 2002.

[65] Ivan E Sutherland. A head-mounted three dimensional display. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, pages 757–764. ACM, 1968.

[66] David Talbot. Graphics Chips Help Process Big Data Sets in Milliseconds. *MIT Technology Review*, 2013, October 18th.

[67] James J Thomas. *Illuminating the Path:[the research and development agenda for visual analytics]*. IEEE Computer Society, 2005.

[68] Allen B Tucker. *Computer Science Handbook*. CRC press, 2004.

[69] Sa Wang, Zhengli Mao, Changhai Zeng, Huili Gong, Shanshan Li, and Beibei Chen. A New Method of Virtual Reality Based on Unity3D. In *Geoinformatics, 2010 18th International Conference on*, pages 1–5. IEEE, 2010.

[70] Zachary Weber and Vijay Gadepally. Using 3D Printing to Visualize Social Media Big Data. *arXiv.org*, 2014.

[71] John Wesolowski. How To Plan Optimizations with Unity*. `https://software.intel.com/en-us/articles/how-to-plan-optimizations-with-unity?language=it`, January, 15 2014.

[72] Jeremy JD White and Robert E Roth. TwitterHitter: Geovisual analytics for harvesting insight from volunteered geographic information. In *Proceedings of GIScience*, volume 2010, 2010.

[73] Thomas Victor Williams. *A man-machine interface for interpreting electron density maps*. PhD thesis, University of North Carolina at Chapel Hill, 1982.

[74] Song Zhang, Cagatay Demiralp, Daniel F Keefe, Marco DaSilva, David H Laidlaw, BD Greenberg, Peter J Basser, Carlo Pierpaoli, EA Chiocca, and Thomas S Deisboeck. An immersive virtual environment for DT-MRI volume visualization applications: a case study. In *Visualization, 2001. VIS'01. Proceedings*, pages 437–584. IEEE, 2001.