JACOBS
UNIVERSITY

High Dimensional Data Visualization in VR and comparison with other approaches in 2D

by

Zhao Liu

Bachelor Thesis in Computer Science

Prof.Michael Sedlmair
Bachelor Thesis Supervisor

Date of Submission: May 16, 2018

Jacobs University — Focus Area Mobility

With my signature, I certify that this thesis has been written by me using only the indicates resources and materials. Where I have presented data and results, the data and results are complete, genuine, and have been obtained by me unless otherwise acknowledged; where my results derive from computer programs, these computer programs have been written by me unless otherwise acknowledged. I further confirm that this thesis has not been submitted, either in part or as a whole, for any other academic degree at this or another institution.

Signature

Bremen, May.15 2018

Abstract

Visualizing high dimensional data has a lot of challenge, a common approach is to use dimensionality reduction (DR) technique, and then visualize it with 2D scatter plot, interactive 3D scatter plot, or scatterplot matrices (SPLOMs). 3D scatter plot suffers from the lack of depth perception and difficulties of navigating through 3D spaces and become the loser in those 3 techniques. However, there are many reasons justifying the use of 3D scatterplots. Virtual Reality technology (VR) can provide a vivid simulation of a 3D world. With the goal of improving the visualization quality of high dimensional data, I explored and implemented an VR prototype of an interactive 3D scatter plot. I analyze the advantages and disadvantages of VR 3D scatterplot and compare it with other existing technique by conducting an user study.

Contents

1 Introduction

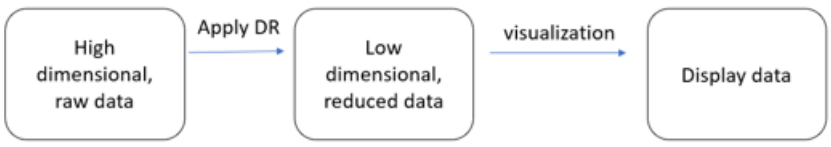
1.1 Background

In the time of big data, there is a need to process high dimensional data to find patterns and hidden structures in the data to use for further analysis. However, challenges of visualizing high dimensional data are well known. With higher dimension, there are more difficulties to visualize it in a regular 2D screen. Also, There is also a high data complexity in multi-dimensional data set, which often combine numerical measurements, images, time, categorical labels, text, geographic location, etc. Feature vectors with tens, hundreds, or even thousands of dimensions at once represent the key challenge of data-driven discovery: the scope of data and the method of processing introduce new opportunities to find connections in data, while at the same time they demand a new set of tools to support the particular qualities of investigating massive data.[7].

Visualization is the main connection between the quantitative content of the data and human intuition, and it can be shown that we cannot really understand or intuitively comprehend the data, especially in high dimension that we cannot visualize in some way. Humans have a remarkable pattern recognition system in our brains, and the ability for science discovery in data-driven science relies critically on our ability to perform effective visual exploration. With the rapid growth of computational science and machine learning, this may be one of the principal methodological challenges for the data-rich science in the 21st century.

1.2 Common Approaches of Visualizing High Dimensional Data

A common approach to visualize high dimensional data is first to apply various dimensionality reduction methods first, then we visualize or low dimensional, reduced data [8]. A great number of dimensionality reduction methods for high dimensional data have been introduced in the recent decade. However, the study[9] has shown that certain methods of dimensionality reduction lead to unavoidable information loss that the reduced data cannot preserve the patterns and structures we want to find. Also, the number of dimensions that reduced data have is also a trade-off: with lower dimensions, it leads to a clear visualization result, while with higher dimensions, it can potentially preserve more interesting patterns and correlations.



1.3 Virtual Reality(VR)

Virtual reality (VR) is a computer-generated scenario that simulates experience through senses and perception. It has been in development for more than 30 years and it recent advances in VR technology give promising of highly intuitive interactions with full 3D

environments relatively straightforward for the first time. [3] It also provides new way of human-computer interaction. The key feature of VR technology is that it includes a head tracking system so the user can navigate in the 3D environment intuitively.

1.4 Overview

This thesis includes a development report on my implementation of a VR 3D scatter plot prototype and explore the use practices of immersive VR as a platform for an interactive, collaborative, scientific data visualisation and visual exploration. I implemented 3D graph-based visualizations, document technical challenges, discuss their technical capabilities and their usefulness as an interactive medium, then conduct a user study to determine the effectiveness of visualisation in VR.

2 Related Work

We review related work on empirical evidence about 2D vs. 3D visualizations and experience and study for implementing VR

2.1 Comparison between 2D scatter plot, interactive 3D scatter plot and scatterplot matrix [1]

Three scatterplot techniques are mainly used for visualizing the reduced data: 2D, interactive 3D, and scatterplot matrices. By focusing on examining the visual separation of clusters after visually encoding the reduced data, this paper analyzed and compared those three common techniques and concluded that in most cases, reducing the data to 2 dimensional reduced data and encoded with a 2D scatter plot gives the ”good enough” results, which means that compare with the 2D, 3D or SPLOMs do not add more cluster separability; Scatterplot matrices are useful and add additional values in certain cases; while interactive 3D scatter plot rarely provide useful results and usually confuse users.

2.2 Problems with 2.5D scatter plot [2]

[10] There are two major problems of a 2.5D scatter plot, one is the facts that when we visualize a 3D scene on a 2D screen, points may overlap and data may be lost in the projection from 3D to 2D. It also makes user hard to tell the distance between two points in the space. Moreover, how to navigating intuitively in a 3D scene is another difficult task to solve. In addition, Study[2] show that using 2D interface results faster at storing and retrieving pages in the display. All of these problems make 2.5D scatter plot rarely useful.

2.3 VR provide depth perception and new interaction[3, 4]

VR has been increasingly used to simulate a realistic 3D scene, one of the reasons is it gives a user the ability to judge the distance of two objects in the 3D world. Several brain

study and cognitive science research are focusing on the distance estimation between two points based on the depth perception provided by VR. A study has shown that users are able to perceive the distances in the right metric order even when only very simple virtual environments are presented.

However, VR has been developed as more as an entertainment tool; most applications are for gaming and movies. Only a few start-ups have VR projects dealing with visualization of high dimensional data. However, most of these products only use 2D plots and makes it fancier. In other words, they simply put more 2D visualization graphs like line charts, bar charts and 2D scatter plots in the 3D space, fails to take the full potential the VR can bring, an immersive 3D simulation, which a 3D scatter plot can truly take advantage of.

2.4 Justifications of the use of 3D scatter plots

Despite the known perceptual problems of visualizing non-spatial, abstract data in 3D [2], reducing to three dimensions and visually encoding with 3D scatter plots is still a frequent practice. [11, 12] A typical justification for the use of 3D scatter plots is that the intrinsic dimensionality of a dataset is likely to be greater than 2; that is, that it would take more than just 2 dimensions to closely approximate the information in the dataset.

A different rationale for the use of 3D scatterplots is that a choice of reducing the original dataset to three dimensions dictates the choice of visually encoding that data using 3D. Also, we are biologically optimised to see the world and the patterns in it in 3 Dimensions.

3 Design and Implementation

Very general idea about the design of this work is to implement a VR 3D scatter plot and analyze if the potential benefit can be realized.

3.1 Tools Used

Thanks to the booming entertainment industry of VR Gaming and VR movies, There are various tools provided for implementing a VR project. For my thesis prototype, I choose to implement on Google Cardboard using Unity.

3.1.1 Google Cardboard

Google Cardboard is a VR platform developed by Google for use with a head mount for a smartphone. Named for its fold-out cardboard viewer, the platform is intended as a low-cost system to encourage interest and development in VR applications. [13] Google Cardboard VR makes a normal iOS or Android smartphone the window to the virtual world. Once inserted into the Google Cardboard, the phone tracks user's head movement, so user can easily navigate around in the 3D scene freely and intuitively. In addition to head tracking, the separated left and right images provide depth perception so that

user can tell the distance between any two points in the 3D scene easily. Together with head tracking, this 3D effect creates an immersive experience for simulating a 3D sense.

Interacting with VR app is easy and intuitive. It consists of two interaction methods. The first and simplest is gaze-based interaction. Users simply look at what they want to select and hold their gaze on it for a second or two. The second type of interaction is by using a wired or Bluetooth controller connected to the mobile. In this case, a user looks at what they want to select, then press a button to select it. The controller can also be used to act by pausing a video or controlling a character.

3.1.2 Unity

Unity is a cross-platform game engine, primarily used to develop both three-dimensional and two-dimensional video games and simulations for computers, consoles, and mobile devices. It uses C sharp and JavaScript for scripting and provides solid content pipeline makes it almost effortless to bring in content from a huge variety of tools. There are many advantages of using Unity3D for creating VR applications: First of all, it natively supported VR. By importing the Google cardboard SDK into the Unity Project, I can easily make the scatter plot in 3D and export/compile it as a Google VR application. Second of all, it makes the application very accessible by providing an excellent balance of ease of use and power and supports more than 21 platforms (PC, Console, Mobile, Web) with considerably good performance.

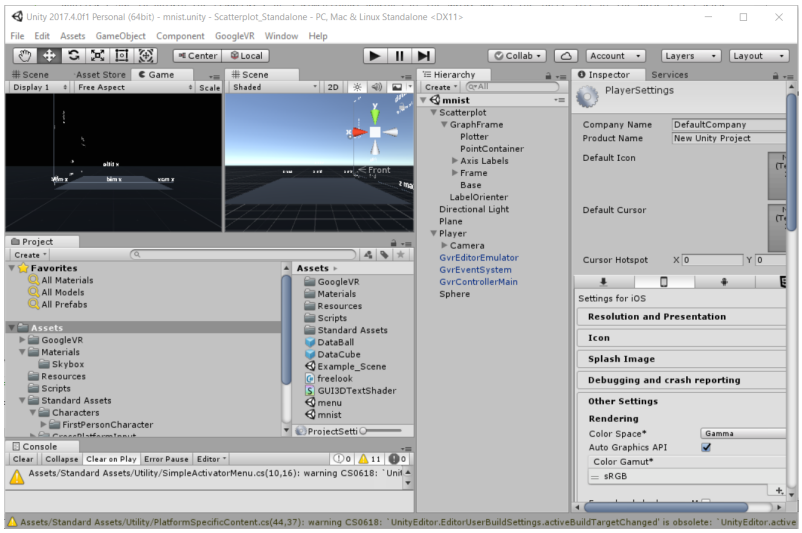


Figure 1: Unity Development Interface

3.2 Camera Movement and UI

A major obstacle of visualization of 3D scatter plot is the camera movement, or viewpoint transition, which requires more complexity than an equivalent 2D visualization. Camera position in 3D consist of two 3D vectors: the position vector represent the physical position in the 3D scene, the rotation vector represent the direction the camera is facing at. The initial idea of implementing the camera movement is to have a controller in hand for

alternating the position vector, and tracking head movement to alternating the rotation vector. However, there are two problem with that. VR Controller are normally designed to navigate on a 2D surface, to walk forward, backward, left and right. In the beginning, I limited the camera to only move on the 2D surface I created under the 3D scatter plot. This limited the view angle to only from bottom to up, did not satisfied the intension of being able to look at it from every angle. It has also been suggested to rotate the 3D scatter plot while the user stay still, but failed to find a way to translate our head movement which intuitively represent the rotation vector of the viewpoint to the rotation vector of the object.

Finally, inspired by the free look camera that are widely used in games, I decided to implement a camera movement based on viewing angle. I lock the viewpoint on the surface of a sphere where the middle of the sphere is the 3D scatter plot we want our camera to focus at. By changing the viewing angle, user can easily rotate around the scatter plot and choose the preferable angle to look at. I also added a reset viewpoint button in case user feels lost while rotating around the scatter plot.

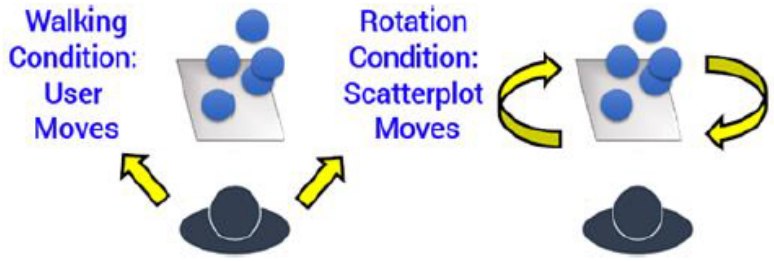


Figure 2: Walking Camera vs Rotation Camera

moving through 3D environments is a high cognitive load task [8]. Shovman et al. [9] even found that restricting degrees of freedom of movement can increase performance for some tasks in 3D. walking condition; rotation condition

3.3 Data set

With the goals of examining the visual separation of clusters after visually encoding the reduced data, the dataset I used to visualize is the MNIST database of handwritten digits, because its clear classification. It use each pixel of the image as a feature, and our high dimensional, raw data is in a vector space of 784 dimensions. First I applied PCA to reduce the initial dimensions from 784 to 50. Then applied t-SNE to obtain the reduced, 2 or 3 dimensional analogs of the data. due to the large size of the data set, I used Barnes-Hut algorithm for better performance on t-SNE.

3.4 Dimension Reduction Methods

Methods of dimension reduction provide a way to understand and visualize the structure of complex datasets. It is used to reduce redundancy and variance, improve accuracy and visualize high dimensional data. As mentioned before, in visualization, dimensionality reduction is used to transform high dimensional data to a low dimensional reduced

Figure 3: example of an entry of MNIST dataset

data while trying to preserve as much information and relationships as possible from the original data. There is no one-and-only DR. While trying different DR techniques; there is no one-and-only DR technique that is superior to all others. t-SNE performed very well with untangling our artificial entangled datasets but did not reveal certain structures in real-world datasets, such as adjacent, stringy classes, which were revealed by the linear PCA techniques. Here I want to talk about two commonly used dimension reduction methods used to for visualization of high dimensional data that I also used in my prototype.

3.4.1 Principal Component Analysis (PCA)[5]

Principal component analysis which is also known as Hotteling or Karhunen is the most commonly used classical algorithms for dimensionality reduction. Unlike the metric and non-metric dimensionality reduction techniques PCA tries to preserve the variance of the data than preserving the distance or the global ordering relations of the objects. For a given high dimensional dataset, PCA finds the vectors along which the data has maximum variance [14]. Generally, PCA transforms the data in to a new coordinate system in such a way that the largest variance by any projection of the data comes to lie in the first coordinate , the second largest variance on the second coordinate and so on. PCA is useful when the data lies on or close to a linear subspace of the dataset. Suppose $x \in \mathbb{R}^{N \times D}$ is a matrix whose rows are D-dimensional data points. We are looking for the d orthogonal vectors along which the data has maximum variance. If in fact the data lies perfectly along a subspace of $\mathbb{R}^D$, PCA will reveal that subspace; otherwise PCA will introduce some error.

It optimizes the objective function

$$V = argmaxVar(XV)$$

Where$x \in \mathbb{R}^{N \times D}$, data matrix $V \in DX$ has its columns as the direction of maximum variance.

As PCA is a linear dimensionality reduction it cannot unfold the low dimensional manifolds embedded in to the high dimensional vector space. The power of PCA algorithm has been extended by applying a kernel trick named as Kernel PCA yet it falls short in handing nonlinear high dimensional data.

3.4.2 t-distributed Stochastic Neighbor Embedding (t-SNE) [6]

t-SNE (tsne) is an algorithm for dimensionality reduction that is well-suited to visualizing high-dimensional data. The name stands for t-distributed Stochastic Neighbor Embedding. The idea is to embed high-dimensional points in low dimensions in a way that respects similarities between points. Nearby points in the high-dimensional space correspond to nearby embedded low-dimensional points, and distant points in high-dimensional space correspond to distant embedded low-dimensional points.

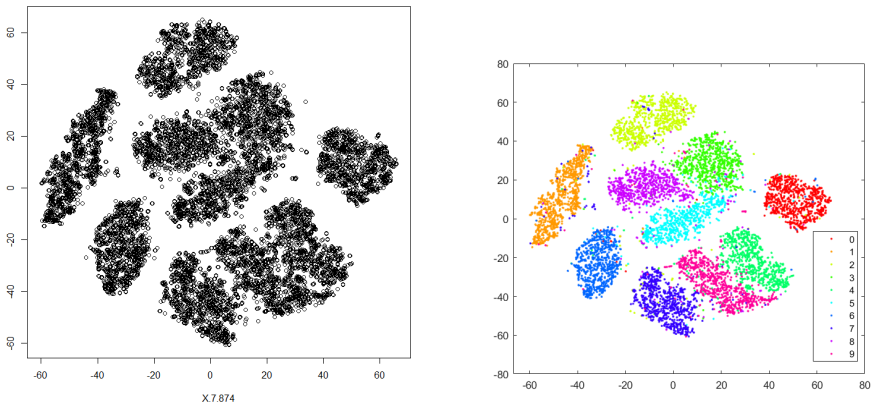
The tsne function creates a set of low-dimensional points from high-dimensional data. Typically, to visualize the low-dimensional points to see natural clusters in the original high-dimensional data.

To speed the t-SNE algorithm and to cut down on its memory usage, tsne offers an approximate optimization scheme. The Barnes-Hut algorithm groups nearby points together to lower the complexity and memory usage of the t-SNE optimization step. The Barnes-Hut algorithm is an approximate optimizer, not an exact optimizer. There is a nonnegative tuning parameter Theta that effects a tradeoff between speed and accuracy. Larger values of 'Theta' give faster but less accurate optimization results. The algorithm is relatively insensitive to 'Theta' values in the range (0.2,0.8).

The Barnes-Hut algorithm groups nearby points in the low-dimensional space, and performs an approximate gradient descent based on these groups. The idea, originally used in astrophysics, is that the gradient is similar for nearby points, so the computations can be simplified.[15]

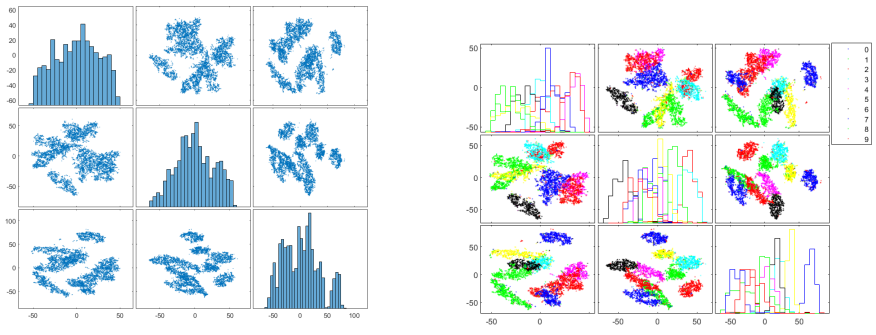
3.5 Visualization Results

Here are the visualization of the 10000 digits from the MNIST data set in 2D scatter plot(Fig.4); scatterplot matrix(Fig.5); and 3D VR scatter plot(Fig.6). I first applied PCA to reduce the initial dimensions of the MNIST data from 784 to 50. Then applied t-SNE to obtain the reduced, 2 or 3 dimensional analogs of the data. due to the large size of the data set, I used Barnes-Hut algorithm for better performance on t-SNE.



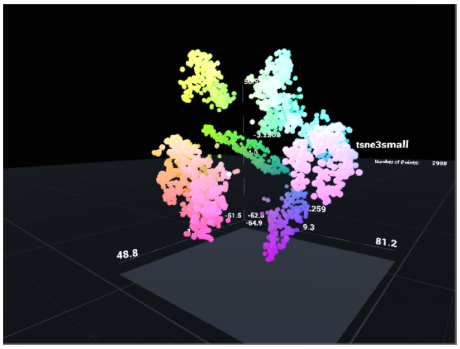
(a) Figure A (b) Figure B

Figure 4: visualization of the 10000 digits from the MNIST data set in 2D scatter plot

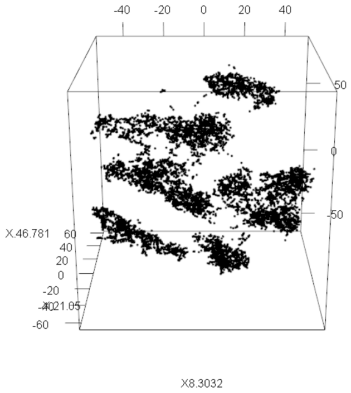


(a) Figure A (b) Figure B

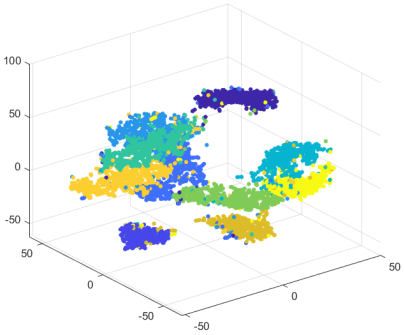
Figure 5: visualization of the 10000 digits from the MNIST data set in scatterplot matrix



(a) Figure A



(b) Figure B



(c) Figure C

Figure 6: visualization of the 10000 digits from the MNIST data set in 3D scatter plot in VR compares to the 2.5D scatter plot

4 Evaluation

To test the effectiveness of my graph to convey information, I developed a user study. The goal of this study was to determine the ability of VR in improving the visual separation of clusters, since the fact that reducing to far fewer dimensions than the datasets intrinsic dimensionality often suffices to make clusters clearly visible after visually encoding. I also want to measure the speed of each user reading data. To do this, we let users experience all 4 of these scatter plots and then ask them to fill out a simple survey and a short feedback.

4.1 Survey Questions

The design of the survey questions are aimed for find out each graphs' visual separation of clusters and the user's time cost. The survey questions are as follows:

- On a scale from 1 to 5 with 1 being the least clear and 5 being the most clear, how clear is the visual separation of clusters clear?
- On a scale from 1 to 5 with 1 being the most uncomfortable and 5 being the most comfortable, How comfortable were the user to navigate around in the 3D scene?
- On a scale from 1 to 5 with 1 being the slowest and hardest and 5 being the fastest and easiest, How much time and effort were consumed when analyze the graph?
- On a scale from 1 to 5 with 1 being the least accurate and the 5 being the most accurate, How accurate is the user's separation of clusters compared to the defined 10 clusters?

4.2 Expected Results

We expected the 3D scatter plot in VR to have significant advantages compared to a 2.5D scatter plot and to have the clearest separation of clusters while viewing a 3D scatter plot in VR is a much more time-consuming task, and it might be difficult for some users to navigate in a 3D scene. We expect 2D scatter plot to provide good enough separations of clusters while costing significantly less time on reading.

4.3 Results

The results of my study meet my original hypothesis of 3D scatter plot having the clearest separation of clusters and a significant improvement from a normal 2.5D scatter plot, according to Figure 6, Users gives almost full scores for the clearness of separation of clusters for VR 3D scatter plot. However, study shows that more than expected user feels uncomfortable and limited navigating in the 3D scene. Based on the feedback, the navigation and interaction of 3D app should be improved: a zoom in zoom out feature is recommended to be implemented, color should be more separable and the size of dot should be smaller.

Moreover, my study shows that there are very different time costs to using the three scatterplot variants of 2D, 3D and SPLOMs because of the significant time cost on interaction. A single static 2D scatterplot has a very low time cost: all of the information is directly visible in one region without the need to interact with the representation or mentally relate the views. A SPLOMs has medium cost: there is no interaction, but a user must switch visual attention between regions and mentally relate the information in different views. The time cost of an interactive 3D scatterplot is high because the user must spend significant time rotating the view to see the structure from different angles in order to see relationships hidden by occlusion in any single viewpoint. My study also shows that these interaction costs in 3D are causing substantially much more time than viewing in 2D scatter plot or scatterplot matrix.

Finally, My study meets the conclusion that among the existing study[1] of the three scatter plot: according to Fig.7, the 2D provides a ”good enough” results compared with 3D and scatterplot matrix. Considering the low time cost of reading a 2D scatter plot and various well-developed tools for it, the 2D scatter plot approach is the fastest and the most reliable way of visualizing our high dimensional data. Even if a set of classes is mixed using one DR technique, trying different DR techniques with different parameter might reveal visually separable class in most cases.

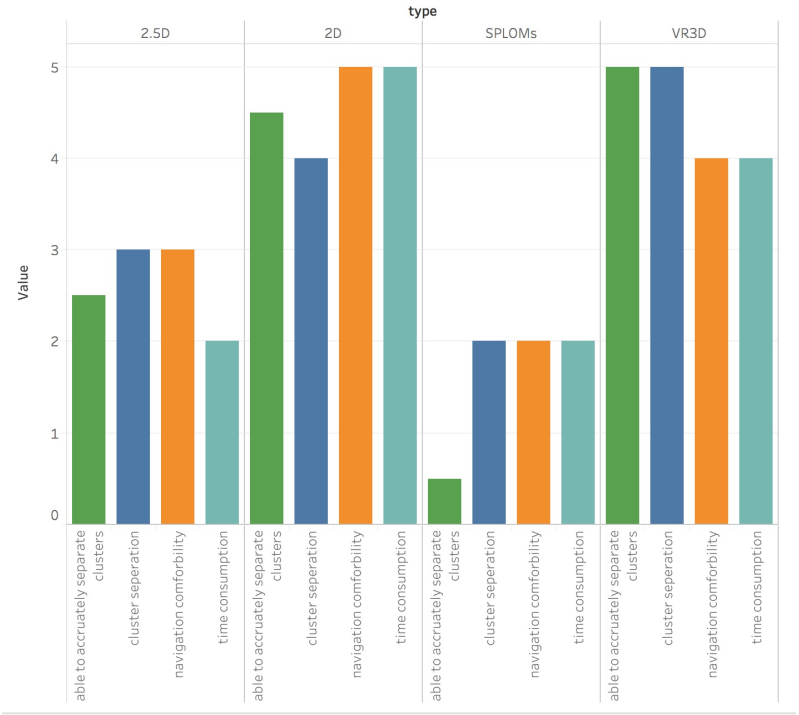


Figure 7: the average score of each survey question

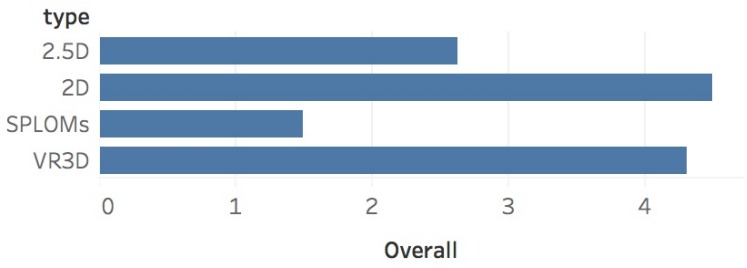


Figure 8: the average score of these four types of scatter plot in my survey

5 Conclusions

There are many challenges in high dimensional data visualization. The common approach is first to apply dimension reduction methods to get low dimensional, reduced data, and then visualize the reduced data using 2D, 3D or scatterplot matrix. 2D is the most promising one while 2.5D scatter plot suffers from multiple problems including difficulties navigating in 3D spaces and fails to tell the distance between two points in 3D. Despite that, 3D scatter plot is justified in certain circumstances. Virtual reality is an exciting field that has the potential to solve the problems of the traditional 2.5D's problem and simulate a vivid 3D world to show 3D scatter plot. In this thesis I implemented a prototype of a VR 3D scatter plot and explore the benefit and disadvantages of this interactive visualization tool by conducting a user study.

According to the study, we can see the significant improvement from 2.5D scatter plot to VR 3D scatter plot by providing depth perception and new ways of interaction. The VR 3D scatter plot is better especially regarding the separation of clusters, is the clearest among these scatter plots. However, it still suffers the problem of all 3D scatter plot: the user takes a significant amount of time on interaction. Also due to the large brain memory need to view the 3D scatter plot, users often feel lost during navigation. In conclusion, 2D scatter plot is still a fast and reliable way to visualize high dimensional data, VR 3D can show a slightly better separation of clusters at expense of time and cost.

6 Future Work

As have already been written, this work is one of the first attempts at exploring the idea of applying VR to visualization of high dimensional data in a 3D scatter plot. There is a huge room for further research. Here are some things that could be done.

- **Improvement of VR technologies**

  Even though a large amount of investment and Research was made in VR area, there are still some issues with our current VR technologies, including the motion sickness that makes people nauseous. An increase of frame rate might be one way to solve this problem.

- **Improvement of VR prototypes**

There are several features that can be added or improved upon my prototype. A zoom-in zoom-out camera; more labels and information; a module to read csv file from the internet so that we do not need to import csv files on the computer and recompile the program everytime we want to change the data we want to visualize would be preferable.

- **More datasets**

  Due to the time limit of this thesis project, I only used one dataset to test my hypothesis. Even though the dataset selected is well known for high dimensional data, more data would most probably make more believable results.

- **Apply More Dimension Reduction Method**

  In my thesis, I only explore 2 of the infinite amount of dimension reduction methods. Even though these two are fit for visualization of high dimensional data, it worth trying to use other dimension reduction technique to test out our results.

References

[1] Michael Sedlmair, Tamara Munzner, and Melanie Tory. Empirical guidance on scatterplot and dimension reduction technique choices. *IEEE transactions on visualization and computer graphics*, 19(12):2634–2643, 2013.

[2] Andy Cockburn and Bruce McKenzie. 3d or not 3d?: evaluating the effect of the third dimension in a document management system. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 434–441. ACM, 2001.

[3] Abdeldjallil Naceri, Ryad Chellali, Fabien Dionnet, and Simone Toma. Depth perception within virtual environments: comparison between two display technologies. *International Journ. on Advances in Intelligent Systems*, 3, 2010.

[4] Claudia Armbrüster, Marc Wolter, Torsten Kuhlen, Will Spijkers, and Bruno Fimm. Depth perception in virtual reality: distance estimations in peri-and extrapersonal space. *Cyberpsychology & Behavior*, 11(1):9–15, 2008.

[5] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[6] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[7] Ciro Donalek, S George 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.

[8] Charles Awono Onana et al. High dimensional data visualization: Advances and challenges. *International Journal of Computer Applications*, 162(10), 2017.

[9] Olga Kurasova, Virginijus Marcinkevicius, Viktor Medvedev, Aurimas Rapecka, and Pavel Stefanovic. Strategies for big data clustering. In *Tools with Artificial Intelligence (ICTAI), 2014 IEEE 26th International Conference on*, pages 740–747. IEEE, 2014.

[10] Gregory B Newby. Empirical study of a 3d visualization for information retrieval tasks. *Journal of Intelligent Information Systems*, 18(1):31–53, 2002.

[11] Ian Doyle, Marianne Ratcliffe, Andrew Walding, Elizabeth Vanden Bon, Michael Dymond, Wendy Tomlinson, David Tilley, Philip Shelton, and Iain Dougall. Differential gene expression analysis in human monocyte-derived macrophages: impact of cigarette smoke on host defence. *Molecular immunology*, 47(5):1058–1065, 2010.

[12] Han Suk Kim, Jürgen P Schulze, Angela C Cone, Gina E Sosinsky, and Maryann E Martone. Dimensionality reduction on multi-dimensional transfer functions for multi-channel volume data sets. *Information visualization*, 9(3):167–180, 2010.

[13] LUCA BIGONI. Creazione di hotspot in ambienti vr 360. 2017.

[14] Johan AK Suykens. Data visualization and dimensionality reduction using kernel maps with a reference point. *IEEE Transactions on Neural Networks*, 19(9):1501–1517, 2008.

[15] t-sne mathwork. https://www.mathworks.com/help/stats/t-sne.html#d119e62044. Accessed: 2018-05-13.

[16] Penn State Big Data Social Science Blog. Beginning data visualization in unity. Blog, 3 2018.

[17] Enrico Bertini and Giuseppe Santucci. Visual quality metrics. In *Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization*, pages 1–5. ACM, 2006.

[18] Cagatay Demiralp, Cullen D Jackson, David B Karelitz, Song Zhang, and David H Laidlaw. Cave and fishtank virtual-reality displays: A qualitative and quantitative comparison. *IEEE transactions on visualization and computer graphics*, 12(3):323–330, 2006.

[19] Enrico Bertini, Andrada Tatu, and Daniel Keim. Quality metrics in high-dimensional data visualization: An overview and systematization. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2203–2212, 2011.

[20] M Sedlmair, Matt Brehmer, S Ingram, and T Munzner. Dimensionality reduction in the wild: Gaps and guidance. *Dept. Comput. Sci., Univ. British Columbia, Vancouver, BC, Canada, Tech. Rep. TR-2012-03*, 2012.