Visualising Origin-Destination Data with Virtual Reality

Functional prototypes and a framework for continued VR research at the ITC faculty

Author: J.K.H Theuns, s1617745

Supervisor: dr. Y. Engelhardt

Critical observer: prof.dr. M.J. Kraak

Word count: 19,250

Abstract:

This report documents the work done by the author in collaboration with researchers at the ITC faculty in Enschede to develop several functioning virtual reality (VR) movement data visualisation (DV/MDV) prototypes, and to develop a holistic, user centered framework for continued virtual reality data visualization (VRDV) research at the ITC. The project broadly follows a design science approach; with literature and state-of-the-art reviews being performed in parallel with iterative prototyping and user testing and requirement analysis. The project reaches several conclusions and recommendations for the ITC including a recommendation for sustained research in the field of VRDV (specifically VR movement data visualization) and development of a data visualisation→ VR roadmap tailored to the ITC's specific workflow, empowering stakeholders to implement and test the VRDV research questions they develop.

Acknowledgements:

The author would like to expressly thank all members of the MoVis workgroup, specifically Yuri Engelhardt, for his continued and consistent academic and personal support throughout the process of this graduation project; Menno-Jan Kraak, for his critical feedback and as a source of inspiration; Luis Calisto, for his technical support and critical feedback, and for Yuhang Gu, Stanislav Ronzhin, Ieva Dobraja and Barend Köbben, for their academic support and guidance, inspiration, collaboration and critical feedback. Special thanks to Erik Bosman of Recreate B.V for his technical support during the initial stages of prototype development.

Table of Contents

Introduction

Context:

Visualising data can be a useful way of communicating and understanding the dynamic relationships of things and their environments. Movement data in particular is very useful to visualise as almost all of humanity's most pressing problems have elements of movement. Ice sheets move and change over time, economic meltdown propagates from place to place over time, as do infectious diseases, animal migratory patterns and human migrations. The movement of things, be that people, animals or goods has been visualised since Harness produced flow maps of people and goods through Ireland in 1837 (fig.0) (Robinson, 1955).

Movement can be broadly defined as either continuous (describing the continued movement of elements through space) or discrete, A → B (describing the start, via and end points of elements over space). This discrete movement is known as origin-destination data, and it is on this kind of data on which this paper will focus.

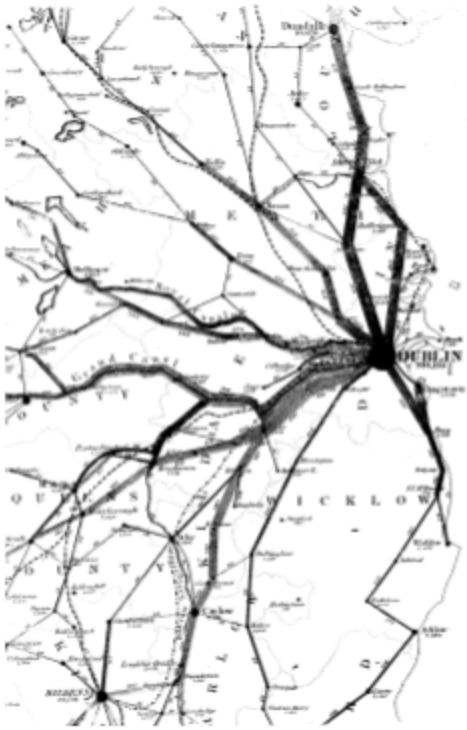

Figure 0: Harness' flow map of Dublin.

As movement data sets get ever larger, spurred on by developments in connectivity, remote sensing and GPS, the challenges in visualising these data sets grow also. Pressing to discover new and more effective methods of visualising this kind of data will play a useful role in understanding (and perhaps solving) the issues outlined above.

This explosion in movement data is accompanied by the recent emergence of high quality virtual reality headsets on the consumer market. VR opens new opportunities to interpret and manipulate digital information in a way much more similar to how we interpret analogue, that is, "real life" information. The headsets do this by tracking the movements of our heads, using this information to constantly update the screens in front of the eyes, with remarkably low latency (1ms). High-end VR headsets and peripherals such as the HTC Vive and Oculus Rift, can track headsets and controllers with 6 degrees of freedom (DoF). Cheaper alternatives such as Samsung's Gear VR and Google's Cardboard only track 3 DoF, but are bringing VR to a wider audience. As this new medium is released, many companies and individuals are racing to create VR games, "experiences" and tools.

Challenges

At the ITC, a faculty dedicated to geo-information science and earth observation within the University of Twente, a small workgroup nicknamed *MoVis* has taken this challenge of

visualising large scale movement data upon itself. A subgroup is exploring the hypothesis that using 3D may allow for the visualisation of larger datasets than what is possible using just 2 dimensions. This hypothesis is twofold: the first claim is that the 3rd dimension gives visualisation designers a new spatial channel with which to visualise complex continuous, quantitative variables before resorting to less effective channels such as colour. The second is that using the 3rd dimension will "hack" the visual system, allowing users to easily distinguish between overlapping and intersecting flows, vital to interpreting flow maps like in figure 0.

This research explores this hypothesis further by incorporating virtual reality systems such that visualisation designers and researchers are not limited to faux 3D depth cues such as perspective, shading and interactive views, but can also incorporate stereopsis, convergence, head coupled motion parallax, and familiar size (one to one tracking of head and hands).

This research recognises the temporal and financial limitations of this particular research and so opts to, instead of exploring the hypothesis directly, to employ design science (fig.1) and design methods to both explore the VR movement data visualisation (VRMDV) design space and to create a framework within which *MoVis* students and staff may explore VRMDV more easily and effectively. This is so that, once I leave the faculty, research can be picked up from where I left off. The faculty did not expressly recommend the use or exploration of VR, this is something I recommended after preliminary meetings discussing the current state of research, the current limitations of OD visualisation, and the equipment available at the ITC (HTC Vive, Leap motion controllers).

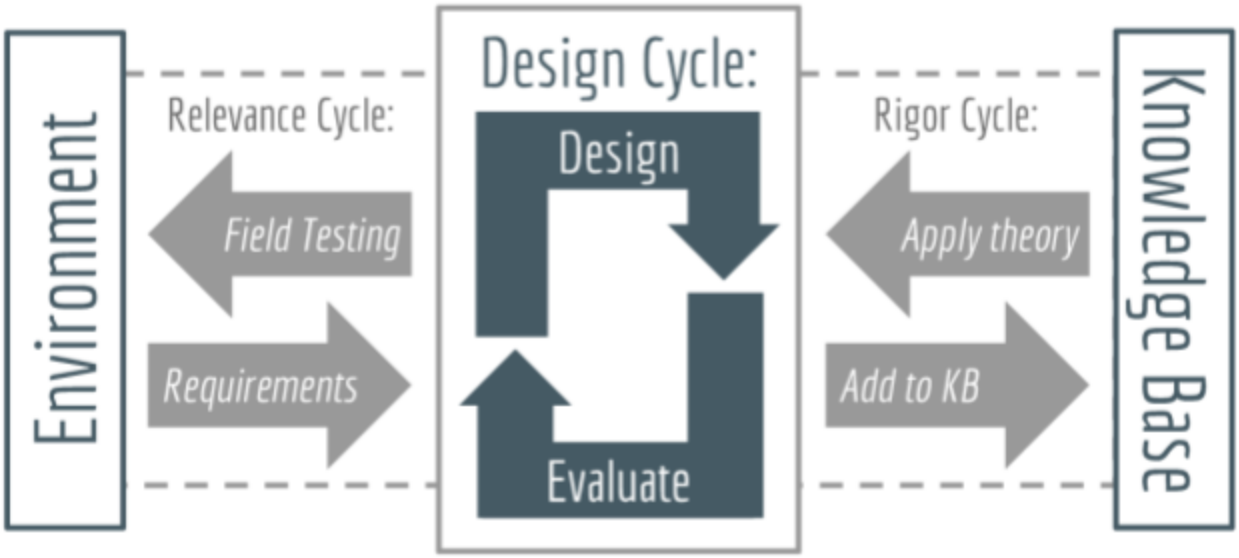

Figure 1: Design Science Approach (A. R. Hevner et al., 2010; Illustration by Author)

Problem Statement:

In the push to gain insights from ever larger OD datasets, we may have reached a limit as to what current visualisation methods can achieve. Utilising current methods and the latest VR equipment, this project will aim to explore the design space of VR OD data visualisation to outline new, promising approaches for visualising OD data. This research will also explore how

researchers at the ITC can make this transition smoothly and easily, so that the promising approaches mentioned above may be explored further in the future. To guide this research, the following research questions have been set:

"What are the possibilities, promising approaches, and potential benefits and drawbacks of viewing origin-destination data in virtual reality as opposed to on traditional monitors?"

- What are the current practices, problems faced, and prospects regarding the visualisation of origin-destination data?
- What are novel possibilities when viewing origin-destination data in virtual reality?
- What are the promising approaches for gaining insight into origin-destination data using virtual reality?
- What are potential benefits and drawbacks of viewing and exploring origin-destination data in virtual reality?
- What factors, besides those in the virtual environment, impact the usability and usefulness of virtual reality data visualisation?
- What factors, besides those in the virtual environment, impact the extent to which researchers at the ITC continue in the field of VR research?
- In what ways could the threshold to virtual reality research be lowered, specifically for researchers at the ITC.

State-of-the-Art and Literature Study

The sections below explore numerous sub questions mentioned in the problem statement. Firstly, the current practises, problems and potential solutions in the field of origin destination data visualisation are explored through literature study. Secondly, the state-of-the-art of VR technology and methods are explored. Thirdly, state-of-the-art VRDV projects are explored and reflected upon. Fourthly, having chosen a specific direction, the benefits and drawbacks of exploratory data analysis in VR are explored. These analyses did not happen in an uninterrupted, chronological stream, rather, they happened in cycles (see rigor cycle, fig.1), with feedback from those at the ITC, and intuitions from the building of the prototypes (see section: Design Cycle) prompting further research.

OD Data visualisation literature study:

This section will focus on deepening understanding of the field of OD data visualisation by analysing the main publications on the topic of the visualisation of cartographic, origin-destination data. The main question this section will answer is "what are the current practices, problems faced, and possibilities regarding the visual analysis of origin destination data ?". This section will first explore the most common (usually oldest) methods and their issues, before exploring and evaluating proposed solutions to the main problems of OD visualisation.

Common methods used, and issues faced when visualising OD data

When creating a visualisation of simple OD data there are three generally accepted methods, with other methods being adapted from other uses. The simplest being an OD matrix (fig. 2). These matrices can be very useful in identifying patterns of flow from origins to destinations. They can represent any number of flows and do not suffer from occlusion (data is not obstructed by other data) and are scalable (can handle any number of origins, destinations and flows). Flow characteristics (distance, count) can be represented by colour (i.e. heatmap) or by numerals. They are limited in that they fail to represent the cartographic, or spatial element. Origins and destinations are usually ordered arbitrarily.

Figure 2: OD Matrix (illustration by author)

Other methods for visualising flow data include chord diagrams. They can show flow counts very graphically, although can suffer from occlusion when there are a large number of linkages.

They also suffer from a similar problem to OD matrices and many other methods of visualising flows (sankey diagrams, bubble charts and parallel sets), in that they cannot visualise cartographic or spatial relationships. There are only two widely accepted methods for visualising simple OD data cartographically: Flow maps (fig. 3), and connection maps (fig. 4).

Figure 3: Flow map (illustration by author)

Figure 4: Connection map (illustration by author)

A famous example of a flow map would be Minard's 1869 map of Napoleon's 1812 march on Moscow (fig. 5); Tufte described it as "the best statistical graphic ever drawn" (Tufte, "Napoleon's march", 2017) and it is widely regarded as a classic. Flow maps are widely used and have many variations, but they also have crucial problems.

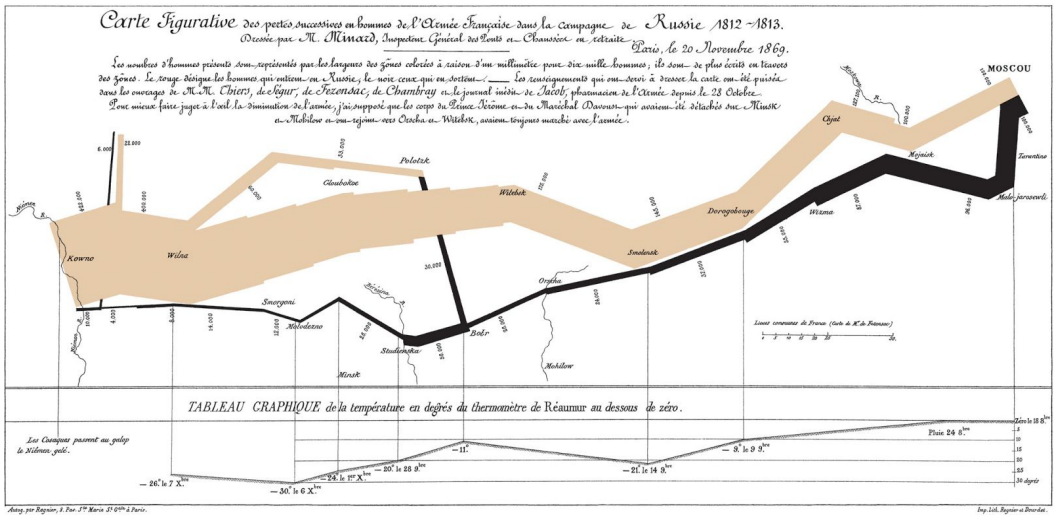

Figure 5: Minard's flow map of Napoleon's march.

Most authors included in this review (below) agree that a large problem when visualising large amounts of geographic network data (including OD) is that overlapping edges (connecting lines) lead to visual clutter (Jenny et al, 2017; Jenny et al, 2016; Zhu, Xi & Guo, 2014; Zhou et al, 2013; Wood, Dykes & Slingsby, 2013; Buchin, Speckmann & Verbeek, 2011; Lambert, Bourqui & Auber, 2010; Andrienko, Andrienko & Wrobel, 2007; Phan et al 2005; Cox, Eick & He, 1996). All present their own methods for solving this problem.

Of these authors, most are also in agreement that the sheer abundance of geographic network data today is staggering, and past methods of drawing each flow map by hand to ensure visual clarity are no longer an option. The drawing of these maps should now be delegated to computers, many articles present novel methods and algorithms for rendering visually pleasing flow maps (Jenny et al, 2017; Zhu, Xi & Guo, 2014; Buchin, Speckmann & Verbeek, 2011; Andrienko, Andrienko & Wrobel, 2007; Phan et al 2005).

There are other, more stylistic, problems as well. Professional cartographers may have similar techniques for depicting direction of flow, count, and geographic elements, however, these techniques are not always used by the ever growing crowd of "amateur" data visualisations. Jenny et al focus on these stylistic choices in their 2016 paper, as do Wood, Dykes and Slingsby (2010).

Potential solutions to the main problems surrounding the visualisation of OD data

Most of the articles analysed in this review present their own potential solutions for the problems of visualising OD data. The potential solutions can be separated into three main groups.

Aggregate data

The first category of techniques, often used when the number of data points is too large, is to cull or aggregate the data. Some articles approach the problem using the technique. That is, they refine or aggregate the data into smaller partitions of which the flows are then visualised.

In their paper, Dykes and Mountain use aggregation methods to turn a large OD dataset into a contiguous surface showing activity density (Dykes & Mountain, 2003). They claim that the technique has benefits over other methods, explaining that this kind of activity density variable can be charted using a single "lightness" variable, allowing other variables to be encoded using hue. Andrienko, Andrienko and Wrobel reflect on this, outlining the limitations of such an approach (2007). They demonstrate that such a "continuous surface map" actually removes the essence of movement from the visualisation. Partially in response to the lack of aggregation methods that succeed in preserving the essence of movement, Andrienko et al outline a framework that aggregates movement data and allows for visual analysis. However, the case studies outlined by Andrienko et al do not demonstrate the effectiveness of visualising truly massive (n > 1000) OD datasets.

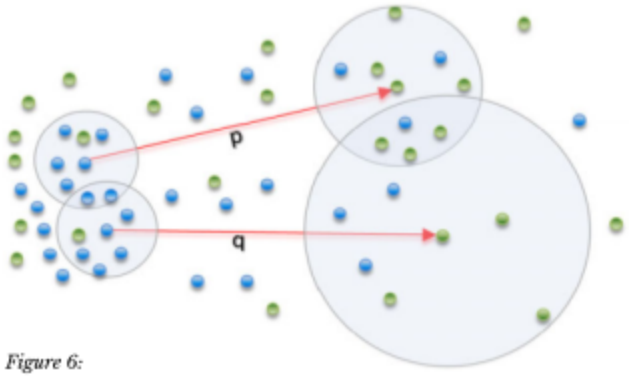

Figure 6:

On the other hand, Zhu, Xi and Guo outline a method for visualising truly massive OD data (2014). Their techniques aggregate flows based on both origin and destination of similar flows and builds upon the limitations of previous techniques.

Instead of clustering based on arbitrary geographic sets (such as states or districts), they create clusters of flows by aggregating flows where the origins and destinations are in similar "neighbourhoods". A neighbourhood defined as a circle with radius *r* around a given origin or destination, such that each point has *k* neighbours within its neighbourhood (fig. 6). The amount of shared points between origins or destinations leads to a distance measure (1 - (shared origin neighbours)/k*(shared destination neighbours)/k)) that is later used to rank pairs of flows. In the case of figure 6, the "distance" between the pair of flows would be 1- (2/7*3/7) ≈ 0.87. If the distance is greater than 1 (either origins or destinations do not share a neighbour) the flows are not considered related. They demonstrate the effectiveness of their technique by visualising 1% of the flows of a taxi data set (fig. 7) and comparing it with the aggregated flows of 95% of the original data using their technique (fig. 8).

Figure 7:

Figure 8:

It is clear to see that their method reduces visual clutter significantly. However, flows are still occluded and the specific case study minimises the drawback of datasets that contain flows of greatly differing lengths (such as commutes).

Refine existing techniques

The second category of techniques looks at how, instead of focusing on the data, the visualisations themselves could be refined to present a solution to visual clutter. For example, in their article on the design principles of OD flow maps, Jenny et al (2016) outline 9 principles to keep in mind when designing flow maps. Their principles are the result of a quantitative content analysis, synthesis of the works of other authors, and a 3 part online survey. Testing was done on the accuracy of user interpretation and user preferences.

1. Number of flow overlaps should be minimised;
2. Sharp bends and excessively asymmetric flows should be avoided;
3. Acute intersection angles should be avoided;
4. Flows must not pass under unconnected nodes;
5. Flows should be radially arranged around nodes;
6. Quantity is best represented by scaled flow width;

7. Flow direction is best indicated with arrowheads;
8. Arrowheads should be scaled with flow width, but arrowheads for thin flows should be enlarged;
9. Overlaps between arrowheads and flows should be avoided.

However, most other articles focus less on design principles and more on how computers can be used to design "good" flow maps. This trend to delegate the drawing of flow maps to computers is becoming a common focus in the field of OD data visualisations. Buchin, Speckmann and Verbeek (2011) demonstrate a technique for the optimisation of computer rendering to more closely resemble hand drawn maps while limiting overlaps. Phan et al (2005) also outline techniques to generate flow maps that resemble hand drawn variants, attempting to reduce the time it takes to produce legible flow maps. Jenny et al also make this computational design a focus in their more recent paper,. Jenny et al outline a method that uses force directed cubic Bézier curves to generate flow maps that adhere to the above principles. Their results are impressive and very applicable. However, they state that their method cannot be used for very large data sets as visual clutter cannot be avoided after a certain number of nodes and links (Jenny et al, 2017).

Edge bundling is also a popular technique used to tackle visual clutter. Edge bundling works by bundling edges with similar origins/destinations (Zhou et al, 2013; Lambert, Bourqui & Auber, 2010). Edge bundling can solve some issues of visual clutter. However, if flows are fully integrated or "bundled", ability to distinguish individual flows becomes a problem. In this way, edge bundling can be seen as a form of visual aggregation. Current literature is focused on how algorithms can bundle flows effectively (Zhou et al, 2013), or on possibilities that arise when bundling edges in 3D (Lambert, Bourqui & Auber, 2010).

Interactivity, being able to highlight and ask for more information on demand, is also a common method used to tackle understandability. Crampton (2002) presents a typology of interactivity in his paper, categorising various methods and their degree of usefulness:

(Low) Interacting with data representation	(Medium) Interacting with temporal dimension
Lighting; Viewpoint; Orientation; Zoom; Rescaling; Remapping	Navigation; Fly-through; Toggling; Sorting and re-expression
(High) Interaction with the data	(High) Contextualising interaction
Database querying and mining; Brushing; Filtering; Highlighting	Multiple views; Combining data layers; Window juxtaposition; Linking

Combining various methods may also present emergent solutions. Andrienko, Andrienko & Wrobel use a multidisciplinary approach to make a case for combining multiple interactive views of data with data handling techniques (Andrienko, Andrienko & Wrobel, 2007). They also stress the importance of combining effective data querying with multiple linked views in order to cater to how human cognition and perception work best.

Modifying existing methods

The third category and least common is to fundamentally change the "normal" way of doing things. Instead of fixing classical node link diagrams, these methods try to visualise OD data using visual encoding that does not occlude itself. Wood, Dykes and Slingsby choose to leave edges behind when it comes to visualising large OD datasets. Arguing that edges of any kind will either aggregate or clutter the visualisation too much, they demonstrate a method that allows the "geographisation" of OD matrices. Instead of using edges to link origins to destinations, they use nested cells (fig. 9). Using this technique allows them to place an OD matrix on top of an existing map. As seen in figure 10, this technique is quite effective at resolving visual clutter, and combining this technique with more conventional flow maps can provide much insight into OD flows. The technique also allows statistical techniques such as the Chi-squared statistic to be applied to the data visually, allowing for exploration of migration relative to the underlying population footprint (Wood, Dykes and Slingsby, 2010).

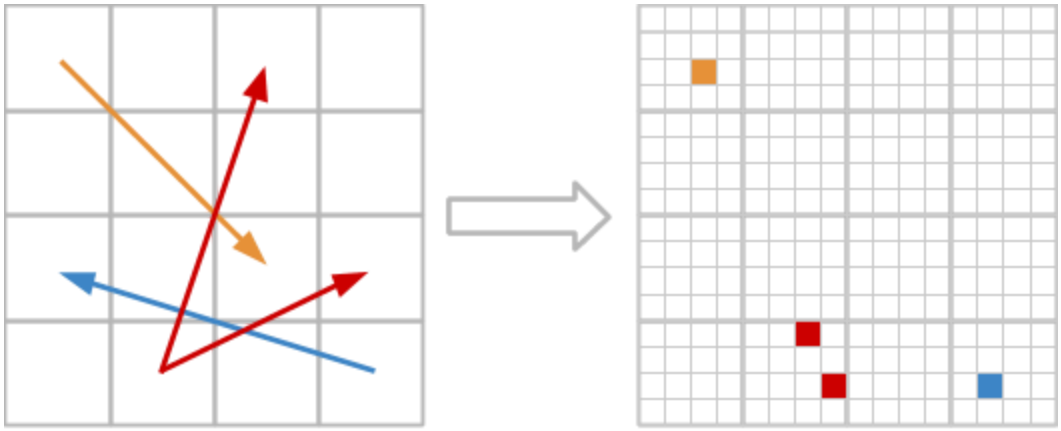

Figure 9: Nested cells solve the issue of occlusion (illustration by author)

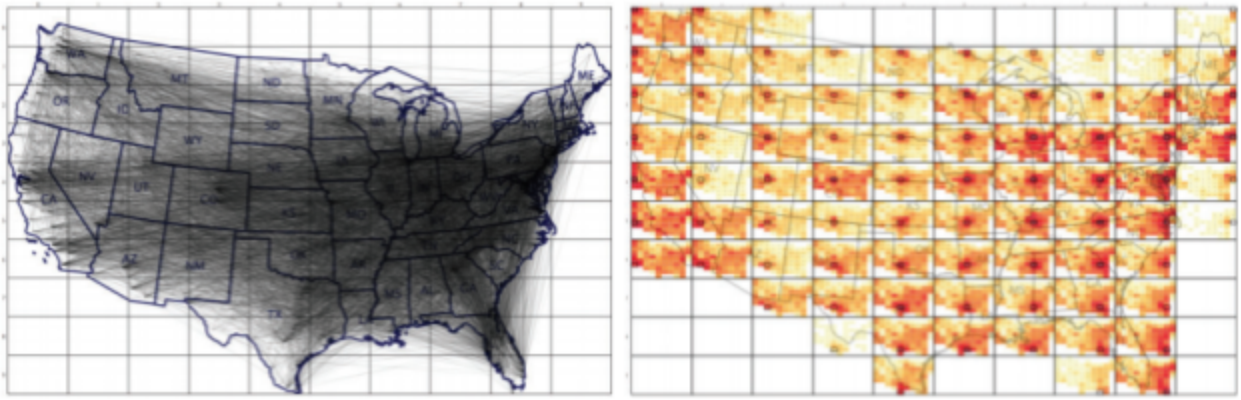

Figure 10: Case study on US county to county migration - OD map example

However, comparing the images in figure 10 one will observe that the OD map aggregates the OD data in a similar way to an aggregated flow map (arguably by a less vigorous method). Whether or not such a technique is as useful as the aggregation techniques demonstrated by Zhu, Xi and Guo would be interesting to examine. Individual county data is lost due to aggregation in any case, and the OD map is arguably less intuitive than a traditional flow map.

Using 3D or 2.5D (faux 3D) can also be a useful way of dealing with the challenges of visual clutter. Lambert, Bourqui & Auber, and Cox, Eick and He focus on how 3D can add to the visualisation of edges in networks. Lambert et al also focus on the effectiveness of edge bundling, while Cox et al focus on various specific 3D data views. In their paper, Lambert et al demonstrate their 3D edge bundling technique. Although their technique demonstrates the rendering of bundled edges using 3D bump mapping, they do not make effective use of the vertical axis (or altitude, in the case of the globe). Cox et al, however, do demonstrate the use of 3D "arcs" for use in visualising flows (Cox, Eick & He, 1996). Their approach aims to preserve geographic context by laying out the networks using multiple, linked, 3D displays. They make the claim that preattentive depth perception of the 3D representation will eliminate the perception of links "crossing", even when viewed on a 2D screen. They make distinctions between global networks, where nodes are positioned on a 3D globe, and arc maps, which lay the geographic information flat on a "map" surface. Their links lay themselves out using a force directed algorithm much like that used in the paper by Jenny et al (Jenny et al, 2017).

In their paper on information visualisation, Ware and Mitchell outline the results of a study where participants were asked whether or not nodes were connected in a network graph (not an OD chart), comparing the impact of various depth cues on accuracy. When various depth cues (stereoscopic vision and motion) were included in the visualisation, answers for two experienced participants were up to 90% accurate when viewing a network of 1000 nodes. Using stereoscopic displays, graphs of an order of magnitude larger could be viewed with similar error rates to those viewed without stereoscopic displays. In the discussion of the paper, Ware and Mitchell conclude that the results support the usefulness of stereoscopic displays in data visualisation. These results are very noteworthy in that they support the hypothesis that VR (stereoscopic displays) may ease visual analysis of complex node - link graphs that present themselves when visualising OD data.

Conclusion and discussion

This review addresses the research question: "what are the current practices, problems faced, and possibilities regarding the visual analysis of origin destination data ?".

It is quite clear that we are living in an increasingly networked world. Access to ever larger datasets means that common methods for visualising OD datasets (such as OD matrices and hand drawn flow maps) are becoming outdated. Common node-link diagrams such as flow maps lead to problems of visual clutter if not designed well or if the number of nodes and links is too large. In the literature studied, three broad categories can be made in how to tackle these issues: Changing the data (aggregation), changing the framework around the visualisations (refine existing techniques) and changing the visualisations themselves (modify existing techniques). For all three categories, focus and research on how the discussed methods may be implemented and optimised using algorithms is present.

With regards to data aggregation, it seems that, given the right circumstances, aggregation based on algorithmic clustering seems to be more promising than data aggregated by arbitrary geopolitical "sections" such as states or counties. When counties and states are used, interesting patterns at smaller scales are missed. Of course, one must always keep in mind what kinds of questions the visualisations are designed to help answer, in some cases aggregations based on these geopolitical divisions are very applicable. However, the development of computationally efficient, algorithmic data clustering has the potential to be a very useful tool.

Aside from aggregation, design and use of the visualisations themselves are also very important. Making use of how we process visual information is essential to making an understandable visualisation. It seems that designing flow maps while keeping these principles of visual encoding in mind is becoming possible to do algorithmically using force direction. Applying some of these techniques in my own work will be useful when creating 3D visualisations. It is also important to avoid making similar mistakes to some of the papers above. Computer graphics have advanced a great deal since 1996 and so, while the papers are useful as examples, design inspiration should be derived elsewhere.

The theory that stereoscopic displays increase accuracy when "reading" node link data is extremely encouraging but should also provide a note of caution. As this effect has such a noticeable effect, when designing VR applications that capitalise and evaluate all the aspects of VR (motion parallax, immersion etc), care should be taken to not only measure the benefits of stereoscopic displays as opposed to traditional monitors, but also these broader aspects.

While I have made my best effort to sample a broad range of well cited papers, this review does not represent a systematic or far reaching study. There are many more sources of information that could provide valid answers to the questions posed. It is important to note that some authors, and the field as a whole, seem to agree that no one method solves the problem by itself. The possibilities that arise when using a holistic approach, combining various methods using multiple linked views, stereoscopic displays, intuitive interaction methods and data aggregation, may provide greater insight than the sum of their parts.

State of the art of VR

This section analyses past decade literature and technology with regards to the main research question: "What are the possibilities, promising approaches, and potential benefits and drawbacks of viewing origin-destination data in virtual reality as opposed to on traditional monitors?". In order to better explore this question, this section will first focus on the current state of VR hardware in general, before looking at specific software examples that relate to data visualisation, and VR development. The last section focuses on industry guidelines for VR development that are relevant to data visualisation.

Current state of virtual reality

"The ultimate display would, of course, be a room within which the computer can control the existence of matter. A chair displayed in such a room would be good enough to sit in. Handcuffs displayed in such a room would be confining, and a bullet displayed in such a room would be fatal. With appropriate programming such a display could literally be the Wonderland into which Alice walked." – ***Ivan Sutherland****, 1965*

A short history of virtual depth

Virtual reality has come a long way since the panoramic, 360 degree paintings of the 19th century. 120x14 meter canvases and viewing platforms such as the one by Mesdag in the Hague (fig. 11) have been replaced by head mounted displays (HMD) and virtual scenery. The state of virtual reality has gradually progressed, incorporating new depth cues incrementally as various technologies progress.

Figure 11: Panorama Mesdag

The first to appear is occlusion, appearing in art as early as 30,000 BC. The use of shading in the depiction of "virtual" scenes, along with aerial perspective (objects in the distance appear hazy) for depicting depth appear with classical art. The use of linear perspective (objects in the distance are smaller) only became formalised during the renaissance, almost 1000 years later (Brooks, 2017). Stereopsis (stereographic images) and panoramas started to appear in the early 19th century.

Virtual depth cues reliant on movement (motion parallax and optical expansion) became possible with the advent of film in the late 19th and early 20th centuries. The incremental combination of these various techniques - along with computer graphics - gradually continued, culminating in the technology of today: VR HMD's. HMD's such as the Oculus Rift and the HTC Vive display binocular scenes with various depth cues generated by computers at a high (90 FPS) frame rate. Their 6 degree of freedom (DoF) tracking capabilities allow for accurate motion parallax and a panoramic effect. Future headsets will aim to increase field of view

(typically 90 degrees in consumer headsets) to that of normal human vision (200-220 degrees), as well as implementing pupil tracked depth of field (already implemented in some experimental headsets but not yet on the consumer market).

Consumer technologies

The virtual reality market today can be separated into 3 groups, mobile, console and PC. Mobile VR uses a simple holder (such as google's cardboard, daydream of samsung's Gear: fig. 12) to mount a user's smartphone, tracking the user's head rotation and generating/displaying the virtual scene. The fidelity of the virtual scene depends on the fidelity of the smartphone used (screen resolution and processor), with some phone manufacturers developing phones specifically for VR (such as google's pixel).

Figure 12: Mobile VR mounts

Console VR such as Sony's PSVR incorporate more processing power and a dedicated HMD to deliver VR experiences that are more immersive, including limited lateral motion tracking (6 DoF instead of 3, as with mobile) and tracked controllers. PC VR is another step up, with the most advanced configurations incorporating submillimeter precision, room scale tracking for headsets, controllers and other tracked peripherals, driven by PC's with state of the art graphics processors such as Nvidia's GTX 1080.

While we are far away from the "perfect display" described above by Sutherland, we are getting closer. Startups focusing on tangible VR are appearing. Omnidirectional treadmills are already on the market and suits that provide tangible feedback are in development.

As the client of this research (the ITC faculty) owns a HTC Vive, I will be using this HMD for testing and design. The rest of this paper will answer the research question with regards to the technical capabilities of the HTC Vive and it's controllers.

Data visualisation in VR.

This section attempts to provide an overview of some concrete examples of VR data visualisation. As it is such a new and emerging medium, the examples offer many different and interesting approaches, all with different end goals. Some applications focus on VR's promise as a storytelling medium, others on its promise for interactivity or collaboration, and others on data saturation (big data visualisation).

Data stories in VR

Users of VR often talk about something known as “presence”. That is, the perception of being physically present in the world you are “seeing” (the VR world) rather than the space you are actually occupying in real space. Capitalising on this effect, some data visualisations illustrate a point using sights we relate to in the real world. *DeathTolls*, an experience created by artist Ali Eslami, uses the allusion of bodies covered in cloths to create gripping and horrifying visualisations of the consequences of war (fig. 13). As the user moves through the 8 minute “experience” they gradually move south east from Europe. Starting with the death tolls of European incidents (Brussels, Paris, Nice), and culminating with a wide vista with 30 towers, each representing 10,000 bodies: the 300,000 victims of the Syrian civil war.

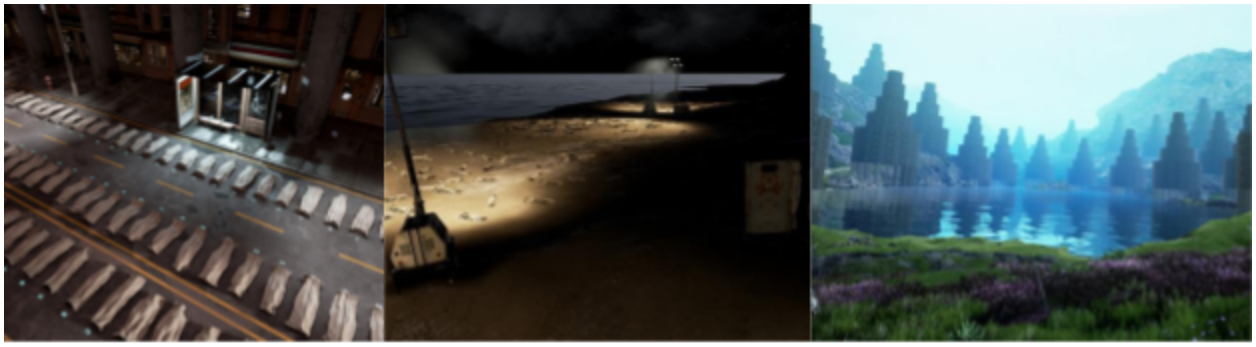

Figure 13: Death Tolls by Ali Eslami

Eslami mentions that traditional methods of data visualisation he tried, such as displaying the totals in contextual displays or adding background information in other ways, such as voice overs, did not do justice to the experience (“The horrors of mass death...”, 2017).

Another data visualisation experience that makes use of immersion to achieve a certain viewer response is one published by the Wall Street Journal about the NASDAQ. Albeit immersive in a very different way, the clever use of height and width of the “path” of the chart conveys the message the chart wants to express quite effectively: The narrow, high path just before the dot com crash depicts the precariousness of the situation very well. The introduction is also effective, guiding the viewer into the experience and providing the contextual information (“Is the NASDAQ in another bubble?...”, 2017).

Interactivity

As high end headsets such as the Vive must be connected to high end PC’s, accurate physics based interaction is becoming a widespread feature of many VR applications. A VR force directed graph application by developer Zach Kinstner demonstrates this technique, using a Leap motion (hand detector) to allow for hands on interaction (fig. 14).

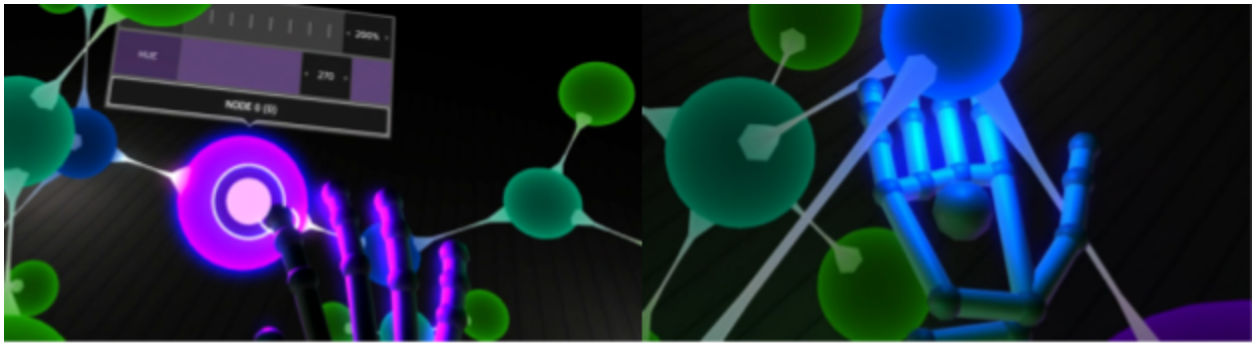

Figure 14: Physical interaction

Other features of Kinstner's visualisation include glowing nodes that allow for intuitive perception of distance between nodes and a user's hands, attractive point and click user interfaces and audio feedback. Physics based lighting helps with immersion, allowing a user to really feel present in the virtual space (Kinstner, 2016). However, there are a few notable drawbacks of the demo. Working in the dark, although visually pleasing, can be disorienting for new users. Physics driven visualisations are also not likely to be able to handle the hundreds or thousands of data points involved in data that suffers from visual clutter. Although, if this physics based interaction and drawing mechanism could be made more efficient so as to work with larger datasets, it might be a valuable tool in increasing immersion and perhaps the effectiveness of the visualisation.

Collaboration

Other start-ups and researchers are focusing on VR's promise to make long distance collaboration more intuitive and immersive. The same potential that VR has for making multiplayer games more immersive, translates into the workspace with "multiplayer" VR data visualisation. The company Virtualitics has made this immersive, collaborative data analysis their goal (fig. 15).

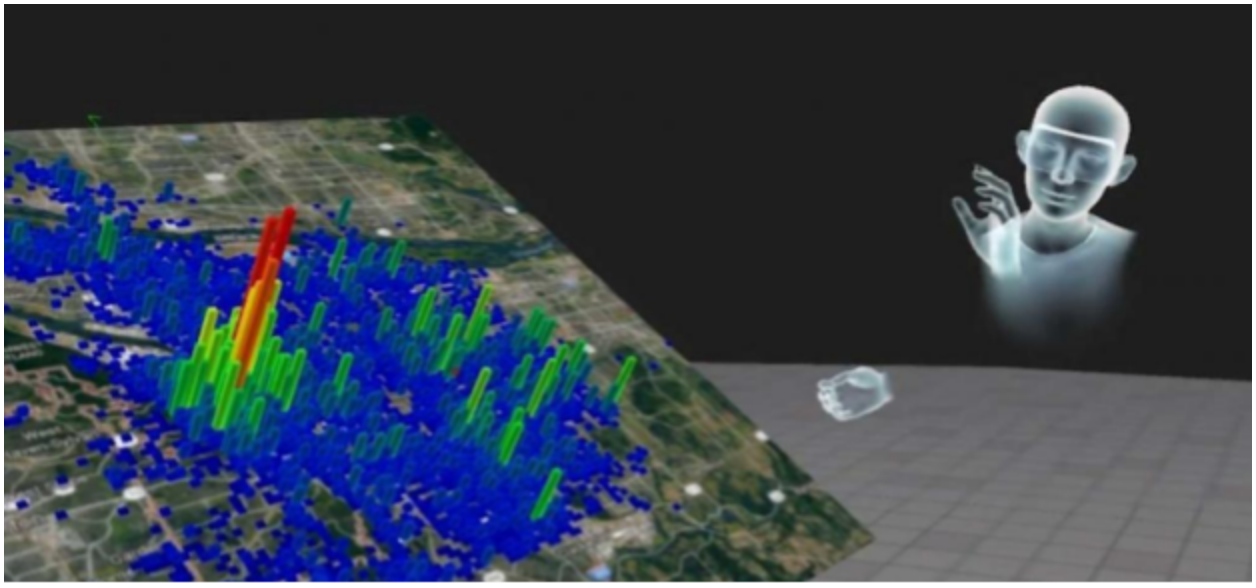

Figure 15: Virtualitics

Virtualitics claim that when data is presented on a screen, users are typically only able to compare data across 5 dimensions, whereas with VR, users can interpret and compare up to 10 dimensions. Virtualitics claim that a shared virtual office space allows users across the globe to collaborate and increases the efficacy and efficiency of VR data visualisation even further (Virtualitics, 2017).

BIG data

The ability to use all the space around a user to support visual information is also prompting a surge in big data visualisation. Companies like Virtualitics (above) and others are focusing on this capability. One that stands out is the project *Masters of Pie.* They effectively visualise the health data of thousands of participants over time, utilising the 3D space to plot more dimensions than would ever be possible with traditional monitors (fig. 16). Their use of the pyramid allows them to visualise 5 dimensions (height above and below axis, breadth, depth and colour) per individual node, organising these nodes in rings around the user offers another two dimensions (height and angle). Rotating docks show contextual information about the selected variables. They won the Big Data VR challenge due to their user centered design process and novel end result.

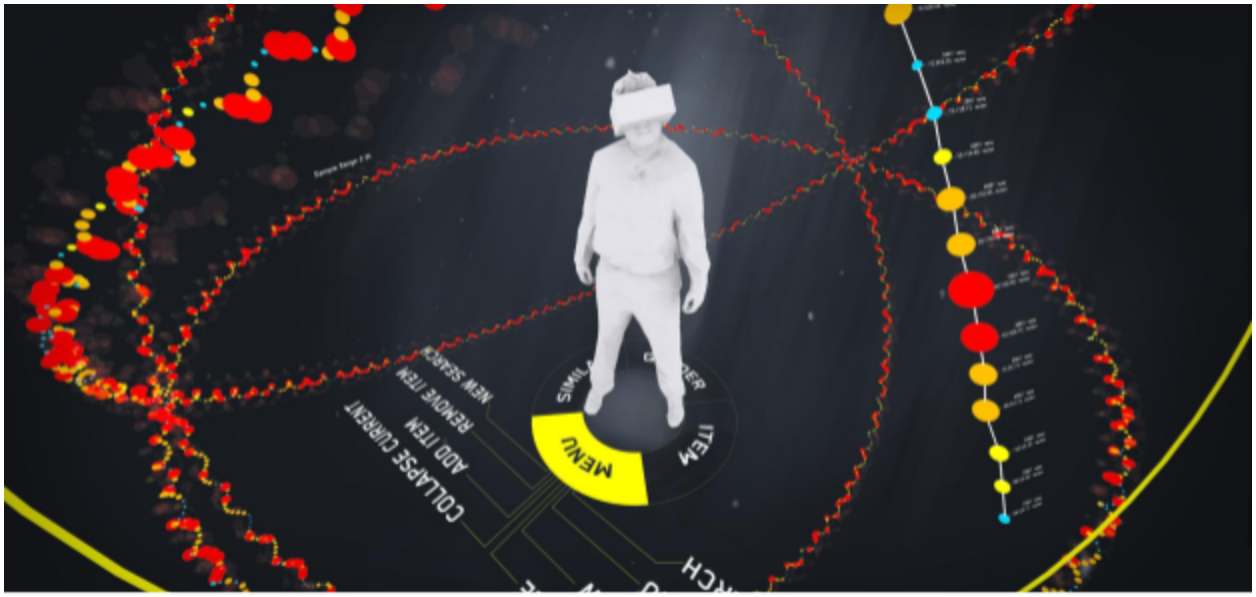

Figure 16: Masters of Pie

Masters of Pi put an emphasis on rethinking the way data is arranged spatially when working in VR. Planar mapping, although very effective on a screen, is not so effective in VR due to the foreshortening of distant data points, making them harder to interpret. Instead of this planar mapping, the group arrange the data radially, with most interface components always in arm's reach, and the data arranged around the user at an effective viewing distance (a virtual 2 meters). Special focus is also put on the ability to interactively filter the data using a dynamic interface, allowing researchers (the target users) to constantly ask new questions of the data and come to new insights on the fly (Masters of pie, “Home”, 2017).

Discussion

The field of virtual reality data visualisation is young and growing extremely fast. It is even likely that the information presented above will no longer be the state of the art before the end of 2017. Advances in hardware allowing even greater graphical (GPU based) and physical (CPU based) processing will push the envelope as to what is possible with brute force methods. Advances in VR hardware may allow for low latency eye tracking, foveal rendering and accurate depth of field by the end of the year. Wireless, desktop VR is already on the market, as well as the possibility to create your own tracked peripherals.

VR data visualisation software is also growing fast. The progress that is made in the coming year will have large repercussions in the whole field. Possibilities are giving way to principles as more designers are learning from the mistakes of the projects discussed in the above section, discovering what is effective and what isn't, working closely with a wide variety of end users to develop very different end products.

If anything, the examples presented above are inspiring. The interesting dark/glow aesthetic and physics based interaction of the force directed graph were ideas I had not yet imagined and will take forward in the design process. The similarities in target users between this project and the MoP visualisation will also undoubtedly have an impact on my design requirements, making sure "meandering" is possible and encouraged in the analysis of the data.

Industry Guidelines for VR development

This section explores guidelines and design manifestos produced by the VR industry. This includes usability issues and design space exploration.

> "Where physical and digital worlds collide, there be dragons."
> - *VR best practices, Guidelines, Leap Motion.*

The design problems thrown up by VR are extremely novel, and in many cases require a complete rethinking of traditional paradigms. For example, as opposed to a 2D screen where visual elements (pixels) are considered to have an x and y position and a colour, this approach does not quite work for VR. VR data visualisation encompasses 2D data visualisation (it is always possible to render a simple 2D data visualisation in VR) but now the positions, sizes, shapes and fundamental qualities of the screen are also taken into question, a matter not usually discussed in data visualisation (except to a limited extent in the design of dashboard environments). A fundamental question of exploring the design space of VR then is to consider not only the positions of visual elements in relation to each other and the canvasses on which they are projected, but also where the user is in relation to the canvases. The whole space must be considered.

Another point is in optics; the HTC Vive's optics are optimised for objects at arm's length, about 1.3 meters. In a presentation on VR, Chu reports that objects are best placed about 10 meters from the user (no more than 20m) for optimal stereoscopic depth perception and no less than 0.5 meters due to blurring (Chu, 2014). Forward field of view is a comfortable 94 degrees, and they report that with little head movement, developers could grab a user's attention within a 158 degree field of view. Maximum field of view without moving the body (max head

rotation) was 204 degrees (fig.17). Using this advice, some functional requirements can be developed.

- Any visualisation or user interface that is being viewed in juxtaposition must be within the 158 degree field of "attention" and would preferably be within the 94 degree forward field of view.
- Any very important elements should be as close as possible to the 1.3 meter "sweetspot radius" depicted below in green.

Looking to the design of workspace environments is a good analogy for this. Users may have a large desk or workbench extending 90 to 150 degrees in front of their field of view. With items used very often (like a keyboard) much closer than objects needed from time to time.

Figure 17: Visual Zones (illustration by author)

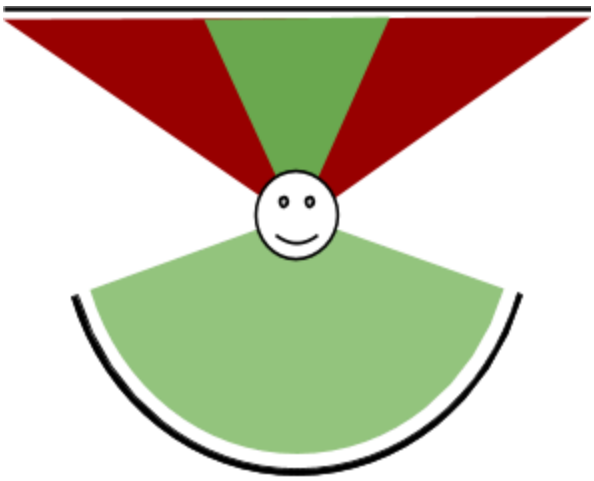

Figure 18: Curved screens and sweet spots (illustration by author)

In this situation it is also important to note the effectiveness of curved user interface canvases. With the relatively small screens (or large screens and small distances) of the real world we assume that this would work well in VR. However, due to the relatively narrow "sweet spot" of VR HMD's, large, flat screens mean that, when one part of the screen is very clear and not distorted, other parts are (fig.18).

In 2014, however, (the date of Chu's presentation) room scale interactivity was not yet possible. This room scale functionality is possible with the Vive, and allows users to actively move around to bring objects they need to interact with or text they need to read at a legible distance. However, while focused on a particular task or some visualisations the recommendations must still apply. Either the user must be able to move objects such that they are able to bring multiple linked views into their own optimal field of view, or visualisations that are linked visually must also be spatially linked.

Users in this scenario can more independently decide what object to bring closer to them, and physically moving through a space may emphasise spatial awareness such as motion parallax when analysing topology or network graphs.

The canvases themselves could be relatively large and far away, such that the movement of the user has little impact on the position of the screen. However this produces issues with the "sweet spot"discussed above. A possible solution maybe be to orient the curved screen to face the user, with the curve of the canvas adjusting such that the focal point is at the position of the user. This would have to be tested however, as I could not find an example of an adaptable curve canvas in existing applications.

The general consensus on text in the VR community at the moment is that it is extremely difficult to pull off. The visual acuity of VR does not allow for the crisp rendering of text we have gotten used to on modern HD monitors. One way to combat this to to use fonts that are especially good at being rendered at small sizes: relatively square sans serif fonts such as Roboto, Lato and Tahoma. Other techniques rely on the method used to render the text, with a favourite for VR being Signed Distance Field texts. They can be rendered at any size without software based pixelation. It remains to be an issue however, and the best way forward may be to limit text to only the most basic tooltips and labels.

In their manifesto on the design principles of hand tracked VR experiences, the developers of Leap motion outline key design guidelines from the VR community as well as insights they have gained from their own usability studies. They make the argument that the physical design of interactive elements in VR should afford particular uses. Handles should appear grabable and buttons pushable. This study of affordances means VR design draws strong parallels with product design ("Explorations in VR", 2015).

How skeuomorphic, or life-like, it would be appropriate to go with VR design is a point of some contention, however. Many objects and interactions in real-life are that way because of the environment in which they exist; VR objects and interactions should be developed similarly, with special attention to the VR environment as a unique medium. Mike Alger, in his video's on VR interaction design, makes the point that many see clipping, or the possibility of going through a virtual object, to be a flaw of VR. He argues instead this property could be exploited, making clipping a useful affordance by making buttons resemble water, something which we go "through" in real life (Thealphamike, "VR Interface Design...", 2015). Flows in movement data visualisation could also be designed in this way, using animation and colour to evoke the idea of liquid rather than immobile virtual bridges.

Another point Alger makes is the importance of VR audio in contributing to presence and how 3D audio can be used as a feedback channel for user feedback. 3D audio, in combination with head tracking could be used to create highly accurate 3D soundscapes. Users would be able to pinpoint where a sound came from in space, this could be extremely useful in VR and is important to consider.

Other important issues include health and safety. In their review on the health and safety implications of VR, Nichols and Patel (2002) mention that cyber sickness is the greatest current health issue surrounding VR. According to their research, prevalence of sickness ranges from 4% to 16%; 94% of the 16% reported symptoms during the first 10 minutes in the virtual environment. Other, more psychological, potential issues including social withdrawal, addiction, and self esteem issues were speculated (especially in the late 90's) but not empirically reviewed. The authors also concede that long term effects, associated with very frequent use that may arise in a professional environment such as the ITC have not yet been examined due to the relative youth of the technology.

One of the ways in which simulator sickness can be mitigated is by improving the objective fidelity of the VR systems and keeping frame rate (the main contributor to cyber sickness) a top priority. Luckily the hardware at the ITC is relatively future proof; the intel i7 6850K processor and GTX 1080 GPU will be able to handle most challenges posed by the researchers at the ITC. When designing the prototype and when developing the VRMDV framework, frame rate and performance must be given a high priority. Another way with which to mitigate cyber sickness is to have visual anchors in the scene that the eye can assume are stable. A horizon line, or similar, can provide this visual anchor.

VR exploratory data analysis

In the above section, a broad look into the possibilities and industry guidelines of data visualisation in VR was presented and explored. In some preliminary analysis of those results and their applicability to the situation at the ITC, I realised that while data stories are an extremely interesting subject to be experimenting with VR, journalism/communication are not what the ITC is focusing on. While both data stories and the exploration of data use visualisations to "translate" data into forms more easily understood by users, the most pressing use case issues (for OD visualisation and data visualisation in general) with relation to the ITC are in exploring data.

Exploratory data analysis (EDA) is a term coined by John W. Tukey in 1961 and refers to the practise of exploring phenomena of interest in data, often using visual methods to highlight unknown patterns. In exploring, users often iterate through visualisations, changing visualisation types and parameters to explore hypotheses relating to the data in question.

In order to explore EDA and the goals and the methods used in EDA, I have explored support for and criticisms of VR EDA in literature as well as the theoretical benefits and drawbacks in relation to OD EDA. Hereafter I explore how the fundamental strengths of VR may be applied in EDA.

Support for and criticisms of VR EDA in literature

There are three main arguments supporting the adoption of VR for EDA applications and two main counter arguments. These will be discussed in more detail below.

- (+) As VR more closely matches our natural way of exploring 3D space, it will allow for more intuitive and effective spatial analysis.
- (+) Effective depth cues allow visualisation designers to layer information, limiting visual clutter and providing another preattentive channel for designers to use in their visualisations.
- (+) As VR provides more space, it allows for the juxtaposition of many large and interconnected visualisations.

- (-) Current VR technology does not provide the resolution nor field of view to visualise large datasets.
- (-) Current consumer graphics cards do not provide the computing power required to simulate truly massive data sets in immersive VR.

Many of these arguments are interrelated, the first and second, and the criticism of the limits of the technology all focus on the phenomena of immersion. In their paper, Bouwman and McMahan (2007), define immersion (as opposed to presence, which is the physiological and subjective response to VR stimuli) as the objective sensory fidelity of a VR system. They argue that high immersion VR systems will allow for greater spatial understanding and reduced visual clutter. Their argument is that as the sensory output of VR systems come to resemble our natural sensory input, our natural optimisation for understanding real 3D space (through systems such as stereopsis, parallax, perspective and occlusion) will help in understanding virtual space. They claim that this increase in spatial understanding could result in greater effectiveness in many applications including exploratory data analysis. They also claim that the increased field of view and display resolution associated with higher visual immersion could have the effect of limiting visual clutter, invaluable when displaying large quantities of visual information. In the results of their study, they found that task performance in spatial analysis was two to three times faster in a high immersion condition, and question responses about spatial relationships, paths and proximity were 3-10 times more accurate.

Ware and Mitchel's observations on increasing the accuracy of complex node-link graph interpretation using high resolution stereographic images seem to confirm these results. As explored in the previous section, their paper outlines the results of a test that showed users could interpret complex node-link graphs significantly more accurately when the graphs were presented stereoscopically. There were yet more accuracy gains when visualisations made use of motion parallax, and with increased resolution. Although the experiments did not include head tracking, by extrapolating the results one could assume even greater benefits when including the six degrees of freedom of a headset like the Vive.

In their paper, Reda *et al* explore the benefits of immersive 3D displays by working with the hybrid reality environment, the CAVE2. The CAVE2 uses 72 stereoscopic panels arranged in a 320 degree, 7.3 meter diameter panorama. This pixel density allows for 20/20 visual acuity at the center of the panorama, with no field of view losses. Each array of LCD's uses multiple computer banks to deliver a total of 22 teraflops of computing power*.

Similarly to the previous authors, Reda *et al* do reiterate the ability of stereoscopic depth to reduce visual clutter. They explore the hypotheses of increased effectiveness and depth as a preattentive channel by working with domain experts to develop cutting edge visualisations using the CAVE2 and other similar systems. In an example, geoscientists at NASA needed to compare 3D topology and 2D graphs depicting chemical properties of certain topographical features in Lake Bonney, an ice covered lake that is one of the places on earth that is most similar to Mars. HR allows for visualisations that can effectively juxtapose such datasets, while allowing for natural interaction in both types. Reda et al claim that the visualisation they developed allowed for a better high level understanding of the lake's topography and how its chemical and biological makeup varies "across the lake and in the water column" (Reda *et al.*, 2013).

Another example was in using stereoscopic depth as a preattentive channel in interpreting the movement patterns of ants. By using depth as a temporal dimension, the visualisation could reveal complex temporal details vital to the research being done. The increased space of the HR display also allowed the researcher to compare multiple trajectories at high resolutions simultaneously.

The last example Reda *et al* discuss was in using immersive 3D to reduce visual clutter in molecular visualisation for nanoscale-materials science. Instead of being limited to 2D representations of the complex shapes and patterns of molecules, scientists could walk around the environment and explore the structures and patterns in the molecules and charge density fields naturally. The high resolution of the CAVE2 gave insight into how complex clouds and volumes can be visualised in the VR systems of the future.

Discussion

The potential benefits of EDA systems in VR mentioned above (reduced visual clutter due to layering, space, depth as a preattentive channel) seem very clearly aligned with the problems faced by OD data visualisation; a main problem of OD visualisation being visual clutter (see previous sections). The increased spatial understanding associated with VR visualisations could also be very applicable in the domain of geoinformation processing, highly applicable given the focus of the ITC faculty.

The main arguments all do seem to revolve around the concept of immersion as both an opportunity and a barrier, with some claiming that there is still a barrier in the technological feasibility of VR applications, opting for hybrid reality installations such as the CAVE2 due to the relatively low resolution and field of view of current VR HMD's limiting their applicability in big data visualisation. While it is true that VR had not reached a level of technological maturity to be perfect for such demanding applications, I would argue that the speed of

---

**Although VR and hybrid reality (HR) are arguably quite different technologies, I would argue that analysis of the design space of today's HR environments is very applicable to the design space of near future VR environments and that Reda* et al's *argument against VR is merely an argument against the current limitations of the technology, not the medium in general. At the time of writing their paper (August 2013), the most advanced consumer VR headset was the Oculus Rift DK1. With a nauseating 60Hz refresh rate, and a 1280 x 800 pixel display, it would indeed be hard to imagine how it would be useful as a data visualisation tool. However, 2 years on, the Vive was released with a 90Hz refresh rate and a 2160 x 1200 pixel display. 90Hz, 4K (3840 x 2160 pixel) displays will be coming to the consumer market by next year (2018). GPU's will have to become faster and cheaper before one could run a 4K or 8K HMD at 90Hz, but GPU's are getting steadily faster and cheaper, as has the software running them (a top of the line GPU today will cost the same as one in 2013, and be 3-5 times more powerful). HMD's today and in the near future will also be able to employ foveated rendering (rendering full resolution only in the direct line of sight of the user using eye tracking), decreasing the loads on the GPU.*

With this steady increase in fidelity and a relatively low price tag ($308 per megapixel for the Vive, compared to $14,000 per megapixel for the CAVE2) one can assume that consumer VR will eventually (possibly quite soon) attain the visual acuity of current CAVE2 systems at a fraction of the cost. Their argument about the limitations of VR is actually a reflection on the

limitations of low immersion VR and, as explored above, it is fair to assume that ultra high immersion VR setups will enter the market very soon.
progression in the field of VR, as well as the vast increases in immersion compared to a traditional monitor (stereoscopic depth, room scale head and hand tracking) warrant exploration of what EDA would look like in VR and how it might function.

Practical benefits of HMD VR in relation to alternative high immersion media such as the CAVE2 need to be considered: Although the 3000 dollars needed to set up a Vive will not seem cheap to most, current alternatives such as the CAVE2 will set you back more than 1,000,000 dollars, along with recurring costs such as the diverse technical expertise needed to maintain such a system. Although useful, the barrier to use is very high. Only very lucky or very rich data scientists will be able to use such systems for the time being.

The CAVE2 does provide 6DoF headtracking, 20/20 visual acuity and the possibility for multiple people to work together with the same visualisation. However, many of the benefits of large systems such as the CAVE2, are also possible with headsets such as the Vive, and, as mentioned above, the immersion of VR systems is increasing steadily: resolution and FOV will become decreasingly important. HMD's require relatively little space to function (a 3*3 meter room is sufficient for room scale applications compared to the 8 x 8 meter space required for the CAVE2), are highly mobile when compared to full room installation, and allow for much higher flexibility with regards to cooperation (users can, but do not have to occupy the same physical space).

As VR technology progresses, the extent to which the mediating technology is a constraint on the mediated experience will continually decline. If /when VR could stimulate the human sensory system in a way such that the mediation itself became "invisible", the medium would no longer seem to play a part in the visualisation; effectiveness in exploring that data would increase proportionally to the effectiveness of the visualisations and analysis themselves rather than the medium with which they are presented.

Design Cycles

The following section documents the creation of the 3 VR prototypes, as well as the field testing that made up the evaluation, and the formulation of a VRDV framework. The first step for the design sections was to condense the information I had learned in the above sections and create an ideal EDA tool. In exploring what the ideal VR EDA tool for OD data might look like, I first came up with a set basic of requirements.

- It must have clear, practical advantages over traditional alternatives.
- It must either be co-designed per function for a particular user, or be broad enough such that a variety of end users have a range of options with which to explore particular OD datasets.
- It must exploit the inherent strengths of VR EDA as explored in the above sections.
- It may be "enjoyable" with some focus being given to a seamless user experience.
- It may employ unique user interaction methods to achieve goals.

EDA is involved with the production (so to speak) of knowledge. To this end, it is highly iterative. Users constantly create ideas of what kinds of patterns and structures might be hiding in the data, comparing different datasets or variables looking for correlations, patterns, outliers. Ideally, when an idea seems to "stick", they would concentrate on denying the existence or significance of said phenomena by again iterating through the visualisation process. They may reach some stronger ideas which they might formally tackle, using statistical techniques to confirm or deny their growing argument. A flow diagram depicting this process is outlined below (fig. 20).

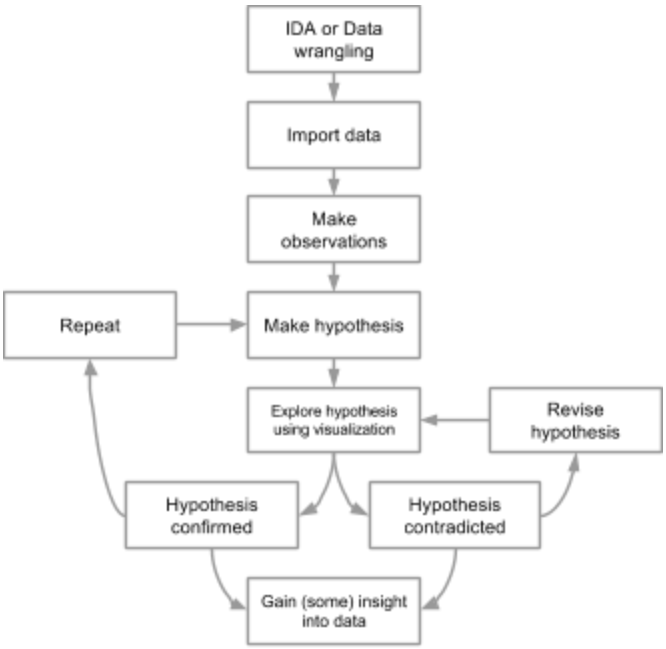

Figure 20: EDA workflow (illustration by author)

It is interesting to note that this kind of explorative analysis resembles the scientific method somewhat. In the natural sciences, hypotheses are tested against nature though experiments, in data visualisations, ideas about patterns and phenomena are tested against the data, often in the form of other visualisations. If the visualisations can be compared to experiments, the EDA tool may be considered a laboratory of visualisations.
In VR, we can move from a metaphorical laboratory of data visualisation to an actual (VR) lab. This lab should be made to take full advantage of the system we are using to project it (the HTC Vive) and should be designed with the ergonomic needs and expectations of the user.
As explored in the section on VR industry guidelines, one way to make the virtual environment accessible is to follow the "visual zones" concept (fig.17). Another idea that was explored, was to let users move interaction items and visualisations around themselves and the room they occupy freely. This would allow users to organize their tasks and the visualisations spatially, which has shown promise in increasing productivity by up to 40% ("Designing VR tools…", 2015). Using the entire room also lets us explore scale, instead of a visualisation being object sized, it could be the room itself, providing an overview while smaller, object size visualisations could provide details on demand. Another benefit VR may provide is the ability to multitask; having multiple, linked views and tasks open simultaneously. This allows users to use information from multiple sources in an analysis, allowing them to explore their hypotheses with greater ease.

First design cycle: The spider

Using the ideas above, I set about creating some requirements for the first VR prototype. The most important requirement was that it had to be a data driven visualisation that was viewable in VR. I chose to use origin destination data representing wikipedia references to and from the page

on data visualisation that I retrieved from a wikipedia crawler (“Medialab tools”, 2017) due to the flexibility of inputs and complexity.

From research it became clear that there were two main frameworks possible to use to create the actual visualisation in VR. The first was WebVR, a browser based tool for implementing VR applications online. The second was to use a game engine like Unreal Engine 4 or Unity. WebVR would have been a great choice for this kind of development. The future of computing in general, browser based applications are becoming popular and may become standard in the coming years. Working online offers the possibility to combine with other web based frameworks such as Three.JS and D3.JS, both common data visualisation tools. Researchers at the ITC are also much more comfortable with tools such as D3.JS. However, WebVR is still quite unstable and very low level, relying on openGL. It only runs on experimental browsers such as firefox nightly and chromium, and, while a strong contender due to the relative simplicity of the data visualisation tools (compared to doing everything manually), it loses to game engines in versatility, flexibility and stability. Engines such as Unity are extremely flexible and offer many more creative options thanks to tools such as the asset store and the plug-ins made especially for use with complex VR applications such as HoverUI and SteamVR. Another (and perhaps the decisive) point was that, at the time, I only had experience with the industry standards (Unity and C#). It is important to note that, while I chose to use Unity myself when designing the VR prototypes, further stakeholder analysis done in the later stages of the project lead to the recommendation to use WebVR for future research at the ITC.

Using Unity as a platform I collaborated with Erik Bosman to create scripts that would allow me to transform the JSON data into visual primitives. The JSON data was in the format:

```
{
	"nodes": [{
		"label": String,
		"x": Float,
		"y": Float,
		"size": Int,
		"level": Int,
		"seed": Bool,
		"colour": Hex-Code,
		"id": String,
	}

		(more nodes)

	],

	"edges": [{
		"index": Int,
		"colour": Hex-Code,
		"id": String: node_id_source -> node_id_target,
		"source": String: node_id_source,
		"target": String: node_id_target,
	}

		(more edges)

	]
}
```

The C# scripts that created the visualisation were the data handler, the visualizer, and the classes of nodes, Bézier curves and edges along with their prefab scripts (premade objects that hold multiple script behaviours and can be reused without changing the scripts that constitute them, comparable to data primitives).

For usabilities sake I created an interface in which a user could input a url leading to a JSON file, or put the name of the file (if it was located in the resources folder). The data handler would then use this information to find and iterate over the JSON text, creating instances of the classes “node” and “edge” that would be stored in an array. If the user then clicked a button “visualize”, the visualizer script would use the lists of nodes and edges would be used to generate instances of the respective prefabs, changing the position, colour and id of the objects to match the data.

To visualize the flows, I used a “Bézier” script that used the positions of the connected nodes and the calculated midpoint (with a vertical transformation proportional to the distance between the nodes) to calculate the positions of points that would make up a segmented Bézier curve. These positions were plugged into a line renderer which was part of the edge prefab. The end result was not pretty, but with help from Unity's SteamVR plugin, it fulfilled the requirements I had outlined for the first prototype: It was completely data driven, it visualized flows and was viewable in VR.

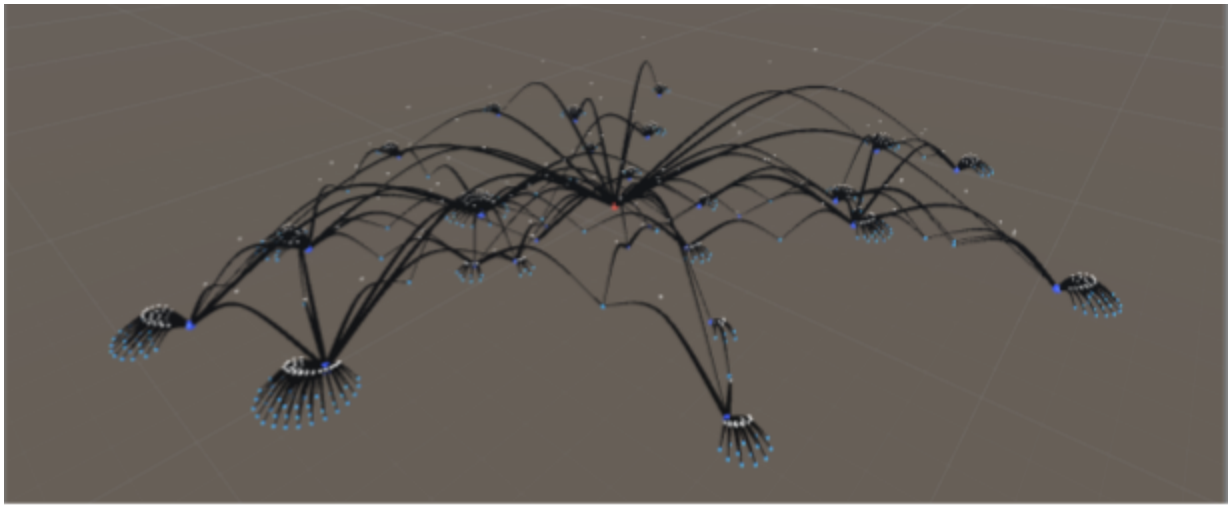

Figure 21: First prototype (screenshot of unity scene)

Evaluation:

To evaluate the prototype, I asked Y. Engelhardt to try on the Vive and think out-loud about the experience. During the session, Engelhardt commented that the visualisation felt better at a scale of about one meter across, rather than the large scale (10m across). He mentioned that he wanted to be able to interact with the points, perhaps with his hands or with the controllers, in order to query them for more details on demand. He also mentioned a kind of movable filter plane, that would look like a window into which a user could examine smaller selections of nodes. During the session, he used the entire space, sitting down on the floor to examine the flows from an alternate angle, commenting that that perspective also had advantages to the “normal” version with flows arching upwards, in that the nodes themselves we no longer occluded by the flows,

making closer spacial relationships easier to discover. The main suggestion was that the next prototype should include some kind of geo-information, such as a base map.

Second design cycle: The Lab

For this and subsequent prototypes, I used a flight dataset given to me by Ieva Dobraja describing flights from Keflavik airport in Iceland. Creating a JSON file from this data with the same (standardised) format described above was difficult. Instead, I used a system of "Triples" where each flight was a single JSON array containing the subobjects: origin, flow and destination. The origin and destination subobjects contained latitude, longitude, airport code, city and country fields. The flow subobject contained arrival and departure times, duration, ongoing connections and number of passengers. To use this data I had to implement a framework to allow location and flow data to be handled by the application. To achieve this, information from classical geographic coordinates had to be mapped to the various local and global coordinate systems to be used in the game engine. Using NASA's blue marble high definition cloudless imagery as the base map, with topographical images as a bump map, I had to convert the lat/long information to the image plane: NASA's imagery is in an equirectangular projection, meaning that latitude and longitude can be mapped to Unity position using the formula:

*pos.x = (longitude + 180) * (image.size.x) / (180) - image.size.x*
*pos.y = (latitude + 90) * (image.size.y) / (90) - image.size.x*

Where: *pos(x,y) = vector position relative the center of the virtual plane*
image.size(x,y) = size of the base map in the Unity world space

The data handler used this data to create nodes for every unique airport code that contained all the relevant location information, and aggregated the number of flights to store in instantiated "edge" objects. Using the "Bézier" script discussed above, with the start and end positions calculated using the above formula and the vertical scale relative to the distance between the nodes, Bézier curves could be generated between the points. With some gradient colouring the following visualisation could be made:

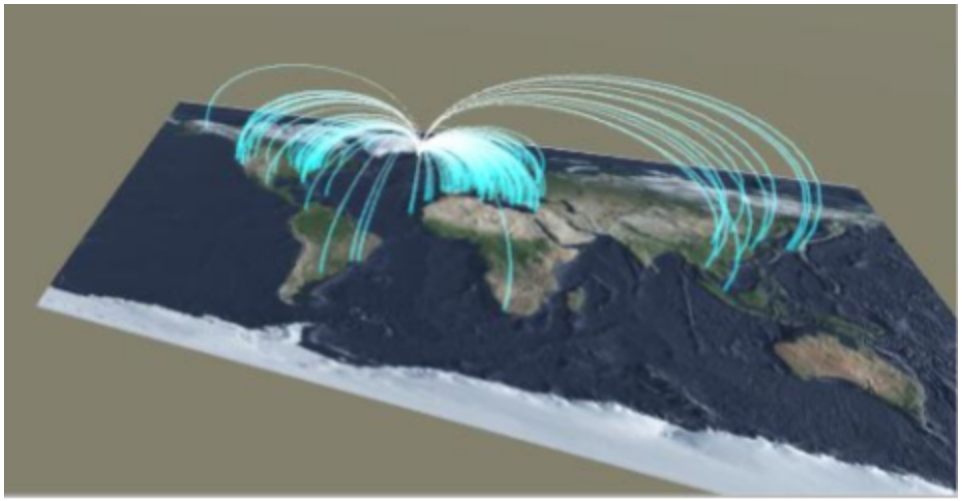

Figure 22: 3D Flow map (screenshot of unity scene)

During a preliminary evaluation, I realized that the planar visualisation gave the impression that flows were predominantly oriented in a vague longitudinal pattern. The patterns of the flows were being influenced by the cut-offs of the projection. To illustrate, picture how the flows would change if the map was split along the atlantic rather than the pacific. The flows would appear to the untrained eye to be originating .

A main benefit of this projection is that it provides a clear overview of the entire map surface. However, due to the challenges outlined above and the unique qualities of VR, an actual 3D globe, not suffering from the problems of projection, may be more appropriate. A globe would not distort the pattern of flows and would give a more accurate representation of exactly how the flights propagated over the earth.

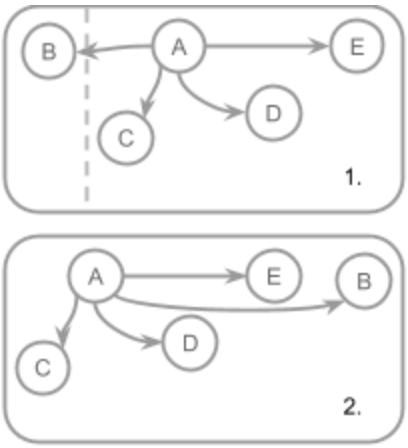

Figure 23: problems with the plane (illustration by author)

Mapping the latitude and longitude information to a sphere was done using the formula:

*pos.x = elevation * Cos ((longitude) * Deg2Rad) * Cos (latitude *Deg2Rad)*
*pos.y = elevation * Sin (latitude * Deg2Rad)*
*pos.z = elevation * Sin ((longitude) * Deg2Rad) * Cos (latitude *Deg2Rad)*

Where: *pos(x,y,z) = vector position relative to the center of the virtual sphere*
Deg2Rad = π / 180°
elevation = radius of the virtual sphere

Using this mapped information to position the nodes, the next step was to implement the flow lines from origins to destinations. Initially, Bézier curves were used (fig. 24), where the position of the intermediary points were directly above the node locations, at an elevation (O → H & D → H) proportional to the distance between the origin and destination (O → D) (fig.24). Although

this technique was effective, there was a critical flaw (flows overlapping and bundling) that was discovered during initial tests (red: fig.24)

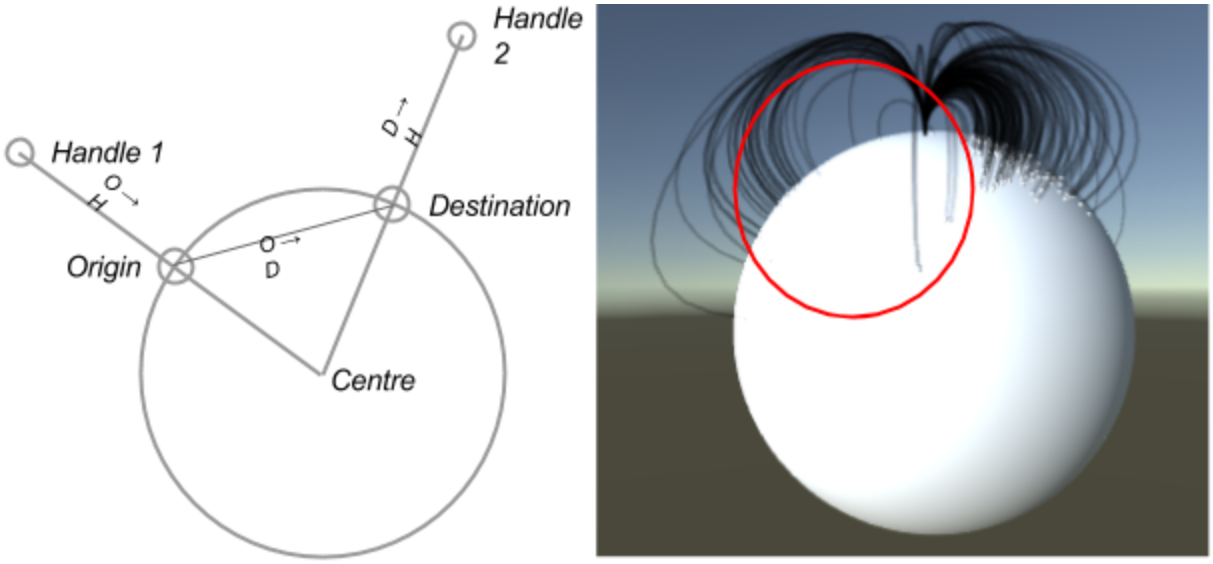

Figure 24: bezier node positions (illustration by author) and screenshot of flows

To explain why this overlapping of flows is happening, let c be the distance O → D, a be the length of the arc between O and D, r be the radius of the globe and θ be the angle made between the origin, globe center and destination in radians.

$$a = r \times \theta$$
$$c = 2 \times sin(\frac{\theta}{2})$$

If r is constant then:

$$c = 2 \times sin(\frac{a}{2})$$

Making the height of the Bézier handles proportional to c meant that changes in θ when close to 180 lead to only small changes in c and thus, only small changes in the height of the Bézier handles, leading to the ugly overlaps seen in figure 24.

I had to develop an alternative method that would allow for greater differences in great arc distance to be visually distinguishable. The new flow paths were created by first calculating the midpoint between the origins and destinations on the surface of the sphere. A game object was created on this point, oriented to the plane made by the origin, midpoint and the center of the sphere. This point became the midpoint of a circle with radius equal to the distance from the midpoint to the origin/destination and aligned to the plane (fig.25). These circles made visually pleasing and distinct arcs that were easy to trace in 3D space (fig.26).

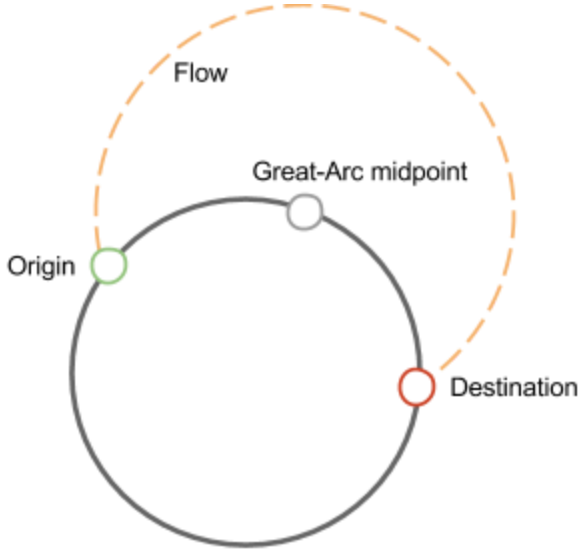

Figure 25: node positions (illustration by author)

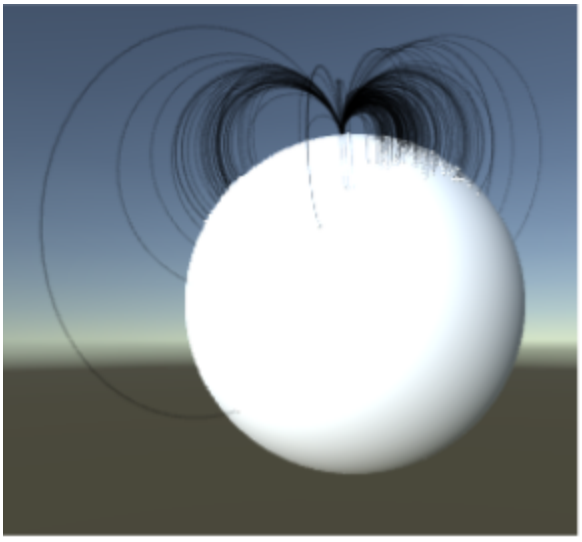

Figure 26: Circular flows (Unity screenshot)

One requirement of creating a globe was that the texture of the base map be mapped correctly to the latitude and longitude. The standard sphere, or UV sphere (a sphere where the texture lines are aligned to the latitude and longitude lines of the globe), would not map the image correctly to its surface because the poles would "pinch" (fig.27). To remedy this, I had to create a sphere that would not "pinch" at the poles. A sphere based on the recursive subdivision of an octahedron allowed for much more accurate spherical texture mapping than the stock sphere that comes with Unity (fig.28). Using the same blue marble composite image from NASA gave the sphere a beautiful surface texture, but this could be improved on in future applications by using a procedural loading system similar to Google Earth VR. I later added atmospheric rendering, optional clouds, a moving sun and night light details to the surface of the sphere, including a starfield.

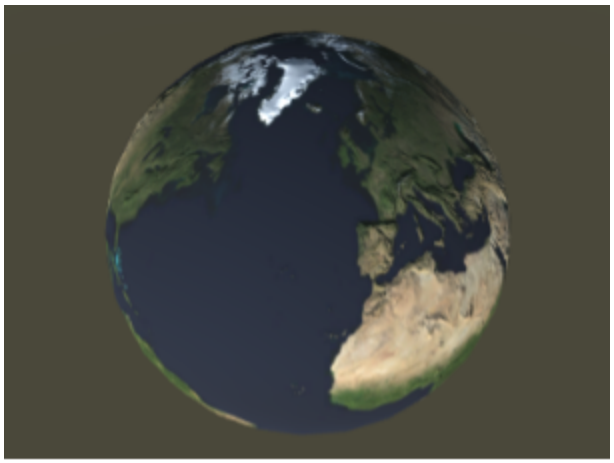

Figure 27: Standard sphere (unity screenshot)

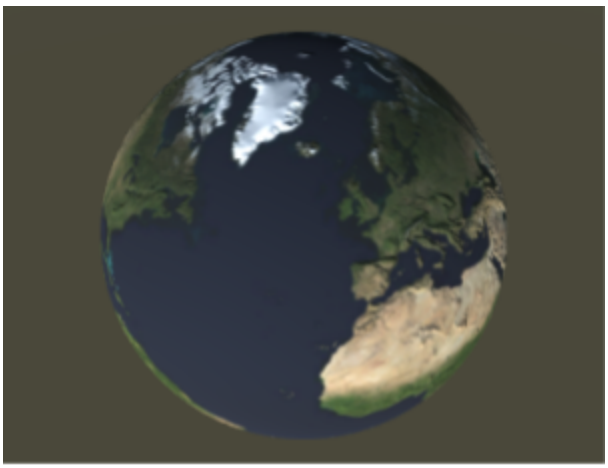

Figure 28: Octahedron sphere (unity screenshot)

One issue that popped up during initial playtesting is that the frame rate would drop from the expected 90 frames per second (fps) to 60 and sometimes 30 or 15fps. As explained in the section on industry guidelines, this would not be acceptable. Numerous methods were employed to combat this. The first was that of optimising the textures and the geometry. The second was to cull the duplicate flows in the data manager rather than the visualizer. The final method, that really helped with start-up speeds, was to load the flows procedurally rather than all at once. This made sure that the memory limit (64000 data points) was not reached. This prevented the frame rate losses associated with motion sickness. The file opens with acceptable speed, and the dynamic loading has the pleasant side effect of acting as a nice loading animation while the visualisation populates itself.

In order to implement the suggestion of emulating liquid flow - discussed in the section on industry guidelines - I wanted to make the flows transparent and animated. To keep overdraw at a minimum while having a transparent effect I used an unlit, transparent texture developed especially for mobile applications. Animation of the flows was achieved by rotating the circles every frame along a perpendicular axis running through the center (like a wheel). To accentuate this rotation, I gave the flows varying width and a gradient of colour (fig.29).

Figure 29: Flow detail (illustration by author)

Now that there were two visualisations in the virtual environment, a way of providing extra information on demand was needed. As discussed in the section on industry guidelines, text is difficult to pull off in VR as the visual acuity of the HMD is no where close to as good as the GUIs we are used to (HD monitors or retina mobile screens). The text needed to be large enough to read and provide enough information. Placing this text on or near the controllers was tested but lead to abandonment. The amount of text (origin, destination, flow, number of flights etc) needed too large a space and ended up occluding the controllers too much, making them harder to use. Placing the text on or over the object being examined also gave design problems. Aligning the the tooltips to the user's viewpoint often lead to text being occluded by flows or basemaps. Eventually, it was decided to place the textual components on their own; details of this method are discussed in the below section on interaction design.

One more question was that of unused space in the virtual environment. The visualisations and text components were looking good, but most of the space was unused. Placing critical information in this unused space would not be prudent (as discussed in the section on industry guidelines, users can only see 94 degrees of the virtual environment at a time). However, leaving this space empty may lead to issues of users not having many visual anchors (during a feedback session, one tester commented that the minimalistic white “hall” that is shown when loading a virtual environment felt uncomfortable and strange, while commenting that the “home-space” (home screen?), full of nicely rendered materials and virtual furniture feels cosy and pleasant). A number of ideas were developed such as leaving the space customisable, a bit like a desktop background, but eventually I settled on turning the whole virtual environment into a vehicle of discovery (of sorts).

The idea was to turn the entire virtual environment into a map, placing users in the locations shown in the dataset using a google-earth like effect to really contextualise the data with virtual (yet much more "real" than a map) environments. To accomplish this, I used the recently released Mapbox Unity SDK. Unfortunately, due to the recency of the release and the fact that most examples were focused on gameplay mechanics, developing this contextualising map visualisation took longer than expected. Using a gaze tracking script, I eventually succeeded in creating a procedurally loaded terrain map (procedural loading to decrease the load on the memory). More details on how users could interact with this map is found in the section below.

Interaction Design

As discussed in the section on interactivity in VR data visualisation , and the sections on VR EDA, VR industry guidelines and *"Potential solutions…",* interaction is extremely important and useful in data visualisation. In the section on potential solutions, an overview of interaction methods and their importance/usefulness is presented. As a medium, VR offers many of the interactions - such as viewpoint, orientation and zoom - natively. Some of the more useful modes of interaction, such as interaction with the data (in the form of filtering and highlighting), and contextualisation (in the form of multiple views, the combining of data layers, window juxtaposition and linking) would be programmed into the virtual environment. The concept of "multiple linked views" was also mentioned frequently during meetings and demonstrations with the MoVis group at the ITC. These ideas, along with the metaphor of the EDA *"lab"* mentioned above under "Design Cycle" would form the basis of the following functional, interaction requirements (from the data visualisation perspective); these were prioritised using the MoSCoW method:

DV Requirement	Priority	Reasoning
Multiple views	**Must**	Andrienko, Andrienko & Wrobel, 2007; Crampton, 2002
Linking	**Must**	Andrienko, Andrienko & Wrobel, 2007; Crampton, 2002; Cox, Eick & He, 1996
Interactive transformation control	**Must**	See page 20
Highlighting	**Should**	Crampton, 2002
Multiple data layers	**Could**	Crampton, 2002
Lighting control	**Could**	Crampton, 2002

From a non functional perspective, the visualisations and their interactions had to be easy to use. Achieving this ease of use would be difficult, however. As VR is such a new medium, most of the users would not have used the Vive or its controllers before. A clear and consistent

interaction paradigm would have to be essential so that users could transfer knowledge learned about one interaction onto another.

To implement these requirements, some kind of event manager and interaction system had to be implemented. There were two possible routes to take here. The first would be to implement a user interface system from scratch using tools provided by Unity. The second would be to use some kind of plug-in. Having experimented with Unity's built in user interface system

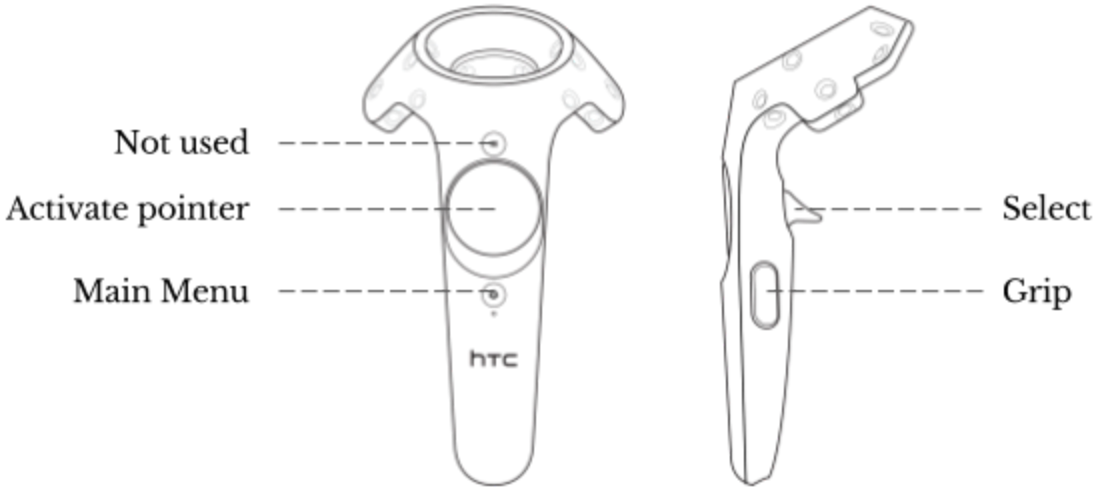

Figure 30: Controller layout (illustration by author, diagram by HTC)

and reaching problems in the first few hours, I realised that the user interface system used by Unity wasn't optimised for 3D interaction. Other systems, such as the hoverUI plugin developed by Aesthetic Interactive, would be more capable for this kind of graphical user interface. In the end, however, I decided to use the VRTK (Virtual reality toolkit) plugin developed by Thestonefox ("Thestonefox/VRTK", 2017). VRTK has a number of advantages over other VR plugins in that it offers solutions for a wide range of interaction problems. Using VRTK, controller behaviours such as on demand pointers, grabbing mechanisms and selections could be implemented with relatively few lines of code, drastically increasing efficiency and allowing for a highly consistent interaction framework.

I settled on an interaction system where users could grip and move objects using the grip buttons on the sides of the controllers (fig.30). The simplest way of doing this would have been to permanently attach the gripped object to the controller during the grip, however, this would allow users to intersect objects. To remedy this a fixed joint with a finite strength was created. This would allow the users to move the visualisations to their liking without permitting intersections (the joints would "break" before an intersection could happen). Gripping an object or visualisation with both controllers would allow the user to scale the object.

The pointer system allowed users to interact with objects they weren't directly touching or intersecting with their controllers. When a user would press the main trackpad (fig.30), a red beam (or ray) would activate and point out of the controller. When the pointer would collide with any object with a collision mesh, it would turn blue, notifying the user that a selection (trigger; fig.30) would be possible. Selection events were activated using the trigger. The consequences of a selection would vary with what was selected.

A table of interactions is shown below. A red box indicates that the interaction was planned but not implemented due to technical limitations, an orange box indicates an interaction that was implemented but not tested and a green box indicates an interaction that was both implemented and tested. Note that all objects with a grip or pointer interaction react to controller/pointer collisions by highlighting, but this feature has not been tested for controller collisions.

Object	Grip (grip button press during controller collision)	Selection (trigger during pointer collision)
Nodes	None	Details shown on text canvas
Flows	None	Details shown on text canvas
Plane, globe	Create a fixed joint between gripped object and controller.	Reverse geocoding on lat/long position of pointer returns country/region name on text canvas
user interface element: buttons	Activate button press	Activate button press
user interface element: sliders	None	Grab slider handle and track movement
Terrain mesh (Mapbox)	None	Teleport to location on mesh
Text canvas	Create a fixed joint between gripped object and controller.	None
Teleport to node location button	Refresh terrain mesh with coordinates of selected node	Refresh terrain mesh with coordinates of selected node

To achieve linking, the data and visualisations were separated, with the visualisations receiving their various data (boolean selection/collision variables etc) from a unified list of instantiated “node” or “flow” data objects stored in the data manager (a model, view, controller structure, fig.31).

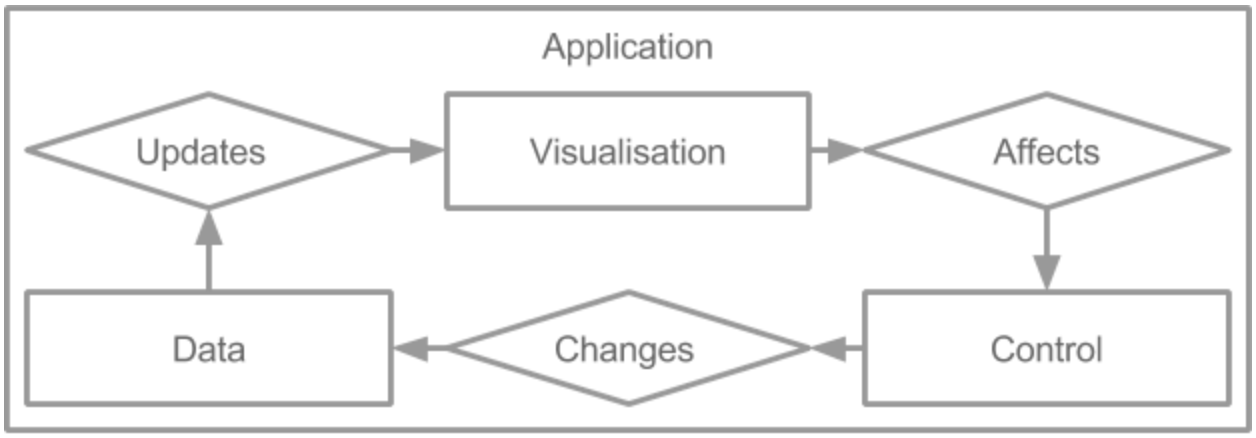

Figure 31: The framework used to propagate changes over all visualisations (illustration by author)

This meant that when a highlight or selection was made, all the affected nodes and flows in all visualisations could be updated to reflect the changes made in real time. Individual characteristics such as how a node or link looked and should respond to highlighting was implemented in the visualisation manager. A custom inspector window was coded to be able to change these variables in the Unity graphical user interface (without having to edit the actual code). Non VR interactions, such as the starting of the application, selection of which visualisations to create and how to populate them, which datasets to use, colours, and the center of the Mapbox terrain mesh were editable in the inspector. In future versions I would have liked to implement a control panel in situ, allowing users to change these variables without leaving the virtual environment.

Allowing for multiple data layers is another element I would like to implement. Creating a graphical user interface to allow filtering of connecting flights (for example) would be relatively simple, a function would iterate over the dataset and disable the nodes and flows in question (disabling would keep the objects in memory but not render them or activate any functions). Unfortunately this feature could not be implemented due to time constraints.

Adjustable lighting was achieved by attaching the main light to a grabbable game object. Although this did not add very much in terms of data analysis (as Crampton's typology suggests - see page 11). It was a small addition that had a large visual and interactive impact: Changing the lighting and seeing the alternating day/night cycles was quite fun and visually pleasing.

Evaluation:

This prototype was evaluated during several focus group sessions, presentations and "think-out-loud" sessions at the ITC geo-visualisation lab during its development. Some feedback was also gathered (although not in an official capacity) during a presentation of the prototype after the final presentations.

The results of the evaluation were, on balance, positive; many of the more critical comments lead to changes being implemented. Indeed, many of the features included in the second prototype were made after feedback from test users. For example, during a test session, many users commented that they wanted the visualisations to be in different positions or with

different sizes. Although this already was an idea to implement, seeing how intuitively people would reach out to try and move or rotate things made the requirement a top priority. Making the flows translucent was also something that was done on recommendation from user evaluation. Initially, the flows were to be translucent, however, after initial tests the overhead in rendering the >200 flows and the complexity in managing the render queue (which items to render first) lead to the idea being abandoned. After the comments from users about the issues of occlusion (flows occluding each other), getting the translucency to work properly also became a priority. The problems above were solved buy using mobile optimised textures that ignored lighting (see above).

Comments did not only lead to implementation of ideas; sometimes they also lead me to discontinue and remove ideas. One example of this was the "sun"; It was a smallish ball that illuminated the globe visualisation. During sessions users commented that the particle systems used to render the sun distracted them and that it was disconcerting if the sun floated through them during its "orbit" of the earth visualisation. After these comments I made the light controller a lot less skeuomorphic: changing its shape to a minimalistic cylinder without the particle systems or the "orbit" feature. Another feature that was removed was the visualisation handles. These were small shapes attached to the visualisations that would *in theory* afford the user to grip and move the visualisations in accordance with the idea of design affordances put forward by Leap (see page 20). In practise, the off center positions of the handles made them hard to use, most notably with the globe visualisation as users would have to move their arms quite far to rotate the visualisation without translating them (changing their positions). Instead, the whole visualisation was used as a handle, allowing the users to change their position in any way without first having to find and hold the handles.

A vast quantity of very sensible feedback was not acted upon, for various reasons. For example, during a quick evaluation session with Y. Engelhardt, he commented that the Mapbox terrain mesh gave the impression of being extremely high up without a surface to stand on. The scale and render distance of the Mapbox terrain will have to be worked on to limit this feeling of vertigo but this could not be implemented at the time due to the very limited support and documentation available about the Mapbox SDK. Not being able to reload the terrain mesh with the locations of the nodes users selected was something that was also missed during the final evaluations for the same reasons; the complexity of the Mapbox SDK and the vagueness of the documentation was difficult to workaround. In general, however, users were positive about the use of the Mapbox SDK to support the contextualisation of the data and the general feeling of actually exploring space, not only data points.

Another frequent comment was that the implementation of other visualisations - such as the OD map by Wood, Dykes and Slingsby (see page 12), or simpler visualisations (such as histograms) depicting routes in order of length, frequency or number of passengers - would have greatly enhanced the overall prototype. These ideas simply could not be implemented from the ground up due to the limited time available. The same can be said for the excellent feedback on details on demand, tooltips and the ability to dynamically change the encoding of a supplementary variable, such as colour, flow width or node size from within the virtual environment.

There was one piece of feedback that acted as a dramatic catalyst; during one of the sessions, one of the researchers at the ITC working with VR commented: "It looks great... but I would never use this". This comment lead to many discussions with the people I was working

with; specifically Luis Calisto and Stanislav Ronzhin. These conversations, in turn, lead to the development of new requirements stemming from insights into the workflows of those working at the ITC. These requirements and the subsequent design cycles are outlined in the next section.

Third design cycle: The Framework

After sessions evaluating and testing the prototypes described above, the comment made about never wanting to use VR for data visualisation kick started this design cycle. I realised that there were two fundamental problems with the approach I had employed thus far: while I had based my work on the literature and made many evaluation cycles to include the envisioned end users into the design stage, the needs of the users *outside* of the prototypes had been missing, and while I had been documenting the project and sharing the process with those at the ITC; I had underestimated the problems the steep learning curve of Unity would present in terms of motivating the researchers to try VRDV.

Unity was chosen because it is an amazing platform with which to develop VR experiences and environments. It is free and had a highly active online support community. As a developer, Unity and Unreal Engine were the only real choices when it came to the question of developing VR experiences and I stand by the decision to use Unity in developing the previous and following prototypes. However, recommending Unity to all the researchers interested in doing VR research was shortsighted. To understand why, I had to look into the workflow of those working at the ITC, the spirit of academia, how Unity doesn't quite mesh into either and how WebVR might.

From discussions and observations at the ITC I came to the conclusion that the workflows of those doing data visualisation t the ITC are very web-based. This is quite easy to understand as a great many of the main data visualisation tools are at least partly or entirely web based. Most notably D3.js; a JavaScript library that has taken the data visualisation community by storm since it's release in 2011; it is currently used on hundreds of thousands of websites. The various reasons for the success of D3 falls outside the scope of this paper but some reasons for its acceptance in the world of academia could be condensed to the following:

D3 uses the web standards HTML, SVG, CSS and JavaScript to function. While this may seem like no big deal, it is. The world of academia has been pushing for the acceptance of universally accepted web standards since the invention of the world wide web. This is mostly due to the benefits that standards can provide in maximizing compatibility, interoperability, safety, repeatability, and quality ("Standardisation", 2017); all of which are extremely important for academia and science. Or to put it in the words of S. Ronzhin, academia is standardisation.

It is perhaps for this reason that the vast majority of workflows at the ITC are based on web standards. Not only D3.js but also three.js, cesium.js, Vega, R-Studio and openlayers all work on the principles of web standards and are widely used at the ITC. Work can be shared easily to anyone with internet access and a modern browser, experiments can be replicated, the mixing and matching of various projects is made easier and security is rarely sacrificed when those standards are used correctly.

Unity, with its range of advantages in usability, power, efficiency and technical support loses out in this field. While it can be used to develop for multiple platforms, it does not do this without significant developer input and expertise and while it can be used to develop web applications, its web builder is notoriously slow. It is primarily a tool for developers, aimed at

creating high fidelity prototypes and products. While it can also be used to make quick explorations, these explorations are hard to share and sharing to be compatible with multiple platforms is even more fraught with obstacles to amateur users. Another important point is that Unity uses C#, a multi-paradigm and powerful, general use scripting language, but very uncommon at the ITC where most use web standards.

In order to fulfill one of the primary requirements - that this project would function as an eyeopener for the ITC to inspire people to incorporate VR into their research - the focus shifted, from creating fully fledged prototypes to creating a data visualisation to VR roadmap to be used by the ITC, as well as generating more, simpler ideas that the researchers could use in future projects, as references and as inspiration. After discussions with L. Calisto and S. Ronzhin, it was determined that further research into WebVR was warranted.

WebVR is a JavaScript API that allows users to access VR experiences from their browsers (mobile or desktop). The idea here is to make VR available to a much wider audience. The main problems with WebVR, and why it has not yet been incorporated into engines such as Unity is that it cannot use threads. Threads allow for the (pseudo) simultaneous execution of different processes by constantly switching between them, rather than executing a process to completion before switching to a new process. This is important for VR applications as (as discussed in the section on industry guidelines) frame rate must stay consistently high to avoid cyber sickness. With single thread processing, such as with WebGL (the JavaScript library that powers WebVR), if something other than frame rendering takes up more than 0.011... seconds to execute (1/90), frames will be dropped, increasing the likelihood of cyber sickness. Both the previous prototypes used threads to load and process the data, allowing for a high frame rate, but this just isn't possible in WebVR as of yet.

Fortunately, this is not a question of which platform one should use to develop fully fledged VR experiences as a VR developer, but rather which platform one should use as a beginner in VR, with some experience in web standards. As WebVR checked those boxes, the next thing to explore was how to develop WebVR applications. As it is based on webGL, the most powerful way to use it would be through a platform such as three.js. This would allow full control over all aspects of the virtual environment, however, three.js is completely abstract (no visual editor) and not an easy to use platform for beginners.

A further search for a more accessible platform lead to the discovery of Aframe . A JavaScript library (web standard) that allows users to code using HTML, also allowing for the incorporation of other JavaScript libraries (more web standards). It uses an entity-component model (rather like Unity) and has a visual editor (also like Unity). Although it is quite limited to creating simple experiences (mostly due to the lack of multithreading), it seemed like an ideal platform to put forward to the ITC.

As an example of its simplicity, replicability and quality, below is a 17 line HTML script that can be opened in any modern browser to allow visualisation of a virtual environment. On a computer, users can move through the environment using the W, A, S, and D keys and look around by clicking and dragging with their mouse. On a phone, the experience can be viewed natively or using a VR peripheral like google's cardboard (see page 16). The scene can also be viewed using desktop HMDs such as the Oculus Rift and HTC Vive by opening the document in a WebVR capable browser such as Firefox Nightly or Chromium (fig.32).

```
<!DOCTYPE html>
<html>
 <head>
  <title>Hello, WebVR! - A-Frame</title>
  <meta name="description" content="Hello, WebVR! - A-Frame">
  <script
src="https://Aframe.io/releases/0.6.0/Aframe.min.js"></script>
 </head>
 <body>
  <a-scene>
   <a-box position="-1 0.5 -3" rotation="0 45 0"
colour="#4CC3D9"></a-box>
   <a-sphere position="0 1.25 -5" radius="1.25"
colour="#EF2D5E"></a-sphere>
   <a-cylinder position="1 0.75 -3" radius="0.5" height="1.5"
colour="#FFC65D"></a-cylinder>
   <a-plane position="0 0 -4" rotation="-90 0 0" width="4" height="4"
colour="#7BC8A4"></a-plane>
   <a-sky colour="#ECECEC"></a-sky>
  </a-scene>
 </body>
</html>
```

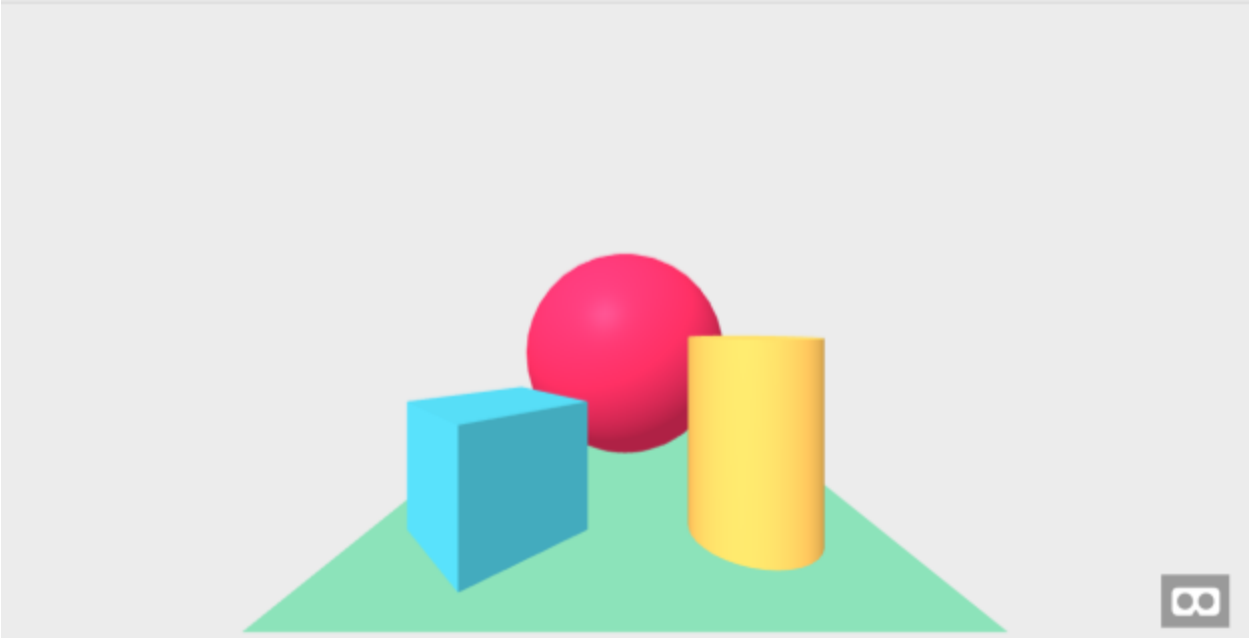

Figure 32: The webVR VE by Aframe (browser screenshot)

Including another JavaScript library such as D3 is done by including the relevant library reference in the header. For D3:

```
<script
src="https://cdnjs.cloudflare.com/ajax/libs/d3/3.5.5/d3.min.js">
</script>
```

Using this method, the D3 functionality researchers are used to can be transferred to VR with a much shallower learning curve.

Evaluation:

During a focus group session at the ITC, I demonstrated the possibilities of VRDV using Aframe. This was done by both trying VRDV examples and by brainstorming the possibilities and limitations for the framework. The reactions could be considered mixed. Of the researchers whose work involved relatively frequent use of JavaScript and HTML, the discovery of Aframe was a breath of fresh air compared to the perceived complexity of Unity and C#. Everyone was relieved to hear that their hard fought understanding of industry standards such as D3 would be useable and extremely useful in developing data visualisation in VR. In general, however, everyone present was made acutely aware that the infrastructure surrounding data visualisation in VR is still extremely weak, compared to the infrastructure surrounding traditional, desktop interfaces, which have largely reached a high level of maturity. It will be a long time before VR data visualisation becomes easy to develop and design, and this ease of use will rely heavily on the (future) emergence of a large industry player, as Spotfire and Tableau have done to the traditional data visualisation market. Until then, VR(M)DV will remain in the realm of data visualisation nd VR experts.

For those experts however, a fair representation of which can be found at the ITC, continuing to use and expand the libraries, API's and plugins developed on web standards will be the key to researching VR data visualisation and movement data visualisation in the future. All present concurred that the framework presented meshed much more fluidly with their own workflow and domain knowledge, and that the framework opened their eyes to the possibilities of VR in an academic context. During the focus group, a number of great ideas in the specific domain of VRMDV came up; one being to implement a space-time-cube in a VR environment. This was quickly implemented in VR (using Unity, for speed) and tested to demonstrate its feasibility and to highlight the importance of a multidisciplinary approach and rapid prototyping when developing for VRMDV.

Fourth design cycle: The space-time-cube

In his paper on the possible applications of the space-time cube in geovisualisation, Kraak presents a clear and useful set of requirements for dynamic space-time cube visualisations (Kraak, 2003). Implementing all of these requirements sadly fell outside of the scope of this project (this particular prototype was developed in a single weekend). The goal of this prototype was instead to serve as a proof of concept with which to inspire the *MoVis* group.

For this prototype I used data from an openstreetmap GPS trace from a street survey done in Comox, BC, Canada. The data was first converted into a JSON format. This JSON file was parsed by a custom data manager that then called a function in a visualisation manager to create nodes with x and y positions relative to the lat/long data, with temporal position being represented by the z axis. A simple mapping function was used to map the lat/long data into

Unity world space while preserving the lat/long spatial relationships. The basemap was generated using the Mapbox Unity SDK with the center of the map being determined using the following formula:

mapCenter.latitude = data.latitude.min + (data.latitude.max - data.latitude.min / 2)
mapCenter.longitude = data.longitude.min + (data.longitude.max - data.longitude.min / 2)

Aligning the zoom level of the map was significantly harder. Ideally, mapping of lat/long data would be handled entirely by the Mapbox SDK. However, limitations in the documentation made this very hard to achieve. A workaround was to scale the data and the map zoom level by hand, but this was (obviously) not ideal.

After the nodes had been positioned, the positions were passed to a line renderer element that drew the path between the nodes. The width of this line was made to vary with measured altitude. In the future, an interface with which users could map line width and colour to variables of their choosing would be ideal. Figure 33 shows the visualisation as it was tested during the focus group sessions. Please note that the visualisation was placed vertically, as a map on the wall, to maximise scale while preventing the need for users to get uncomfortably low down when attempting to view the flow side on.

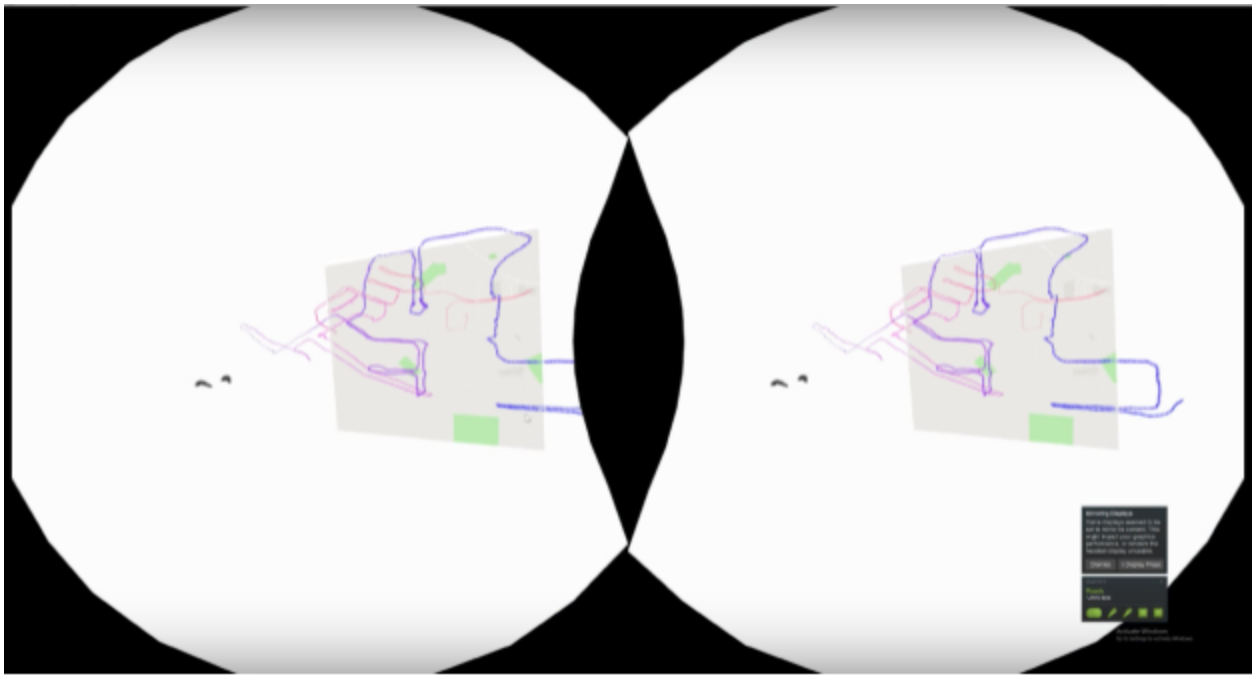

Figure 33: Space-Time Cube prototype (screen-capture still-frame)

Evaluation:

Many of the comments made during the focus group focused on the absence of things mentioned in Kraak's paper (2003). Notably, an axis scale, attribute and 2D views, reference and measurement of nodes in 3D space, the ability to orient the visualisation to the needs of the viewer (without moving the user) and the querying of details on demand. (Some of these elements were implemented in subsequent prototypes but never tested due to instability).

Other comments focused on the usability of the visualisation in VR. Some found the vertical orientation to be quite intuitive, allowing to the traversal of time as a natural traversal of space (walking), others disagreed, mentioning that cultural differences in how maps are perceived (both national and industry specific sub-cultures) may make the “map on the wall” approach more intuitive for some, and the “map on table” or “map on floor”, more intuitive for others. Fortunately, such a preference could be easily implemented in a graphical user interface. Issues such as what orientation options should be given to users in either view (floor or wall) need to be considered however. Other notable were ideas included:

- A tool for drawing a bounding “cube” which could be used to dynamically focus on a particular area of a visualisation
- The generation of reference lines to the basemap and time axis during controller collisions
- Translucent time “planes” (rather than grid lines), appearing in a softly bounded area around the hand-held controllers.
- The possibility to move the basemap through time using a grip interaction similar to that discussed in the second design cycle
- The need for a semi transparent base map, so as to not fully occlude nodes behind the map during the sliding of the base map.

Implementing these ideas using Aframe would be the next step forward for this prototype and would make the prototype much easier to share to receive feedback from all over the movement data visualisation community.

Fifth Design Cycle: The 3D Chord Diagram

During the last focus group, Y. Engelhardt was musing about the globe visualisation in the second prototype and came up with the idea of placing the flows inside the globe rather than around it. The idea was that being inside this visualisation would be a better use of space inside the virtual environment. In order to demonstrate the feasibility of the idea, I quickly (3-4 hours) implemented a prototype that would serve as a proof of concept. In theory, the visualisation would be a 3D chord diagram, with the user in the center (fig.34).

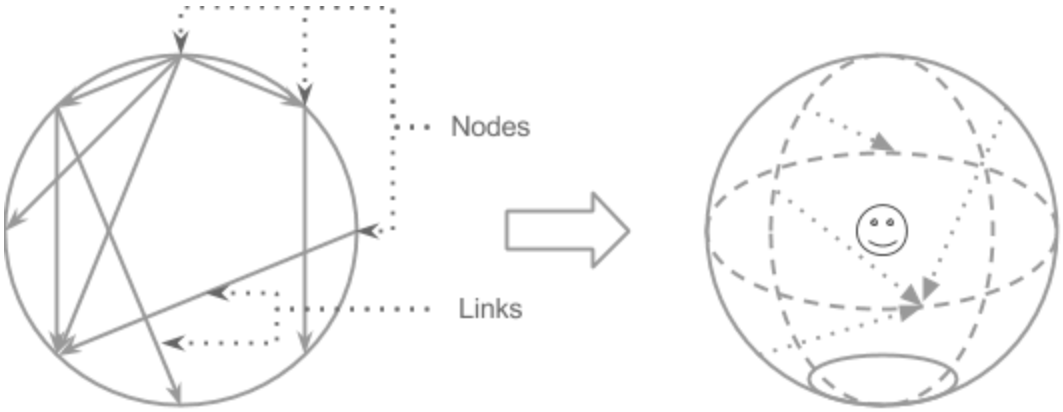

Figure 34: Traditional vs 3d Chord diagram (illustration by author)

The globe view from the second prototype was used as a starting point. Beneath this base map, a standing platform was made, roughly 1.5 meters in diameter. This platform, although perhaps unnecessary, was implemented in accordance with the industry guidelines on visual anchors. As there was to be no visible horizon, something else was needed that would keep the user oriented (see: industry guidelines).

Around the globe visualisation, a second globe, much larger than the first was made. In order to make this globe visible from the inside out, I used a small script that allowed me to inverse the rendering direction of the globe (flipping the normals). The outside globe was textured with the same material as the first, only flipped in the horizontal direction to counteract the flipping of the normals. Using nodes and Béziers much like the original (flawed) Bézier curves of the very first globe visualisation, meant I only had to set the midpoint handles to a negative altitude proportional to the great arc distance in order to render the flows on the inside of the globe rather than the outside. The scale, or by how much the Bézier flows caved in towards the viewer could be adjusted in the inspector (this could also be adjusted by the user in a graphical user interface). Parenting the outside globe to the inside globe and fixing the x, y, and z position of the inside globe while keeping the rotation free meant one could use the inside visualisation to rotate the outside one, adding an extra visual anchor to any user activated movement.

The pointer functionality of the second prototype was kept, allowing users to query nodes to find out more about them. This extra textual detail was presented on a movable text canvas, in the same way as in the second prototype.

Evaluation:

Unfortunately, this prototype could not be tested with all members of the *MoVis* group. However, during presentations of the prototypes after the final presentations, Y. Engelhardt and a number of other (non targeted) users had a chance to try the visualisation. Similar comments were made about this visualisation as with the others, one that stood out was that this visualisation indeed made a better use of the space in the virtual environment to fully immerse a user in the data. The prototype mostly generated a lot of further questions such as:

- What would it be like if the inner globe represented origins and the outer destinations

- Where should other views (if applicable) be placed inside the much smaller space
- What is the ideal size of the outer and inner globes

The prototype definitely succeeded in inspiring various new ideas and research questions through rapid prototyping, and how the “quick and dirty” implementation of those ideas could generate unique insights.

Process evaluation

The following section will reflect on the process of the entire project, focusing on to what extent the project has achieved its stated goals. The goals stated in the beginning of this project were the following:

- To explore the design space of VR movement data visualisation
- To outline new, promising approaches for visualising OD data
- To explore how researchers at the ITC can make the transition to VR research smoothly and easily, such that the promising approaches mentioned above may be explored further in the future

This project has explored the design space of VR movement data visualisation through literature review and the development of four prototypes with varying degrees of complexity and completion. The project has also, though the exploration of the literature and through development of the 3D chord diagram, outlined and tested (preliminarily) new and promising approaches for visualising OD data. This goal may have been better executed through continued iteration of the initial prototype.

The project has also explored how researchers at the ITC could best make the transition to VR research. This process, as well as the other aspects of this project have been guided by a design science approach, building requirements from the knowledge base (literature) and from iterative exploration into the workflows of those at the ITC. This iterative approach has ensured that the recommendations made, specifically the recommendation to explore the Aframe framework, are sound conclusions.

In retrospect, given the volume and complexity of the ideas generated by this project in collaboration with the ITC; this project may have benefited from a larger technical team, allowing for a more rapid turnaround in the creation of new prototypes.

A retrospective argument that could be made is that the researchers at the ITC would have benefited more from this project had I started developing in Aframe from the beginning. While this may be true, the achieved complexity and volume of work made possible by using the framework I was most comfortable with (Unity) likely outweigh the theoretical benefits of working with Aframe from the start.

Conclusion

This project explores the possibilities, promising approaches, and potential benefits and drawbacks of viewing origin-destination data in virtual reality as opposed to on traditional monitors using a design science approach. In doing so, it has generated: A relatively in-depth literature review of the current state of origin destination data visualisation, including the problems and potential solutions in the field. A review on the history, present state and future of virtual reality technology. An exploration of the VR data visualisation applications in the world today. A review of the VR industry's guidelines VR development, specifically how they relate to data visualisation in VR. A discussion into the specific goals of this research from the perspective of the ITC leading to a brief exploration of the support for, and criticisms of VR exploratory data analysis in modern literature. Using this research, four functional prototypes and one meta-framework were designed, implemented and evaluated that explored the research question from a practical point of view. These processes, specifically the implementation and evaluation are explored and documented in detail.

Fundamentally, this project acted to create discussion and a fresh perspective on VR movement data visualisation research at the ITC and I think it has succeeded in that goal, opening the door for future students (of creative technology or otherwise) and researchers to build upon and solidify the processes started here.

Video of the second prototype can be found here:
https://www.youtube.com/watch?v=EcoAOPGzF10

Videos of the focus group sessions can be found here:
https://www.youtube.com/watch?v=ig6qRcuKmoA

References

Andrienko, Gennady, Natalia Andrienko, and Stefan Wrobel. "Visual Analytics Tools for Analysis of Movement Data." *ACM SIGKDD Explorations Newsletter* 9.2 (2007): 38. Print.

Bowman, Doug A., and Ryan P. McMahan. "Virtual Reality: How Much Immersion Is Enough?" *Computer* 40.7 (2007): 36-43. Print.

Brooks, Kevin R. "Depth Perception and the History of Three-Dimensional Art: Who Produced the First Stereoscopic Images?" *I-Perception* 8.1 (2017): 204166951668011. Print.

Buchin, K., B. Speckmann, and K. Verbeek. "Flow Map Layout via Spiral Trees." *IEEE Transactions on Visualization and Computer Graphics* 17.12 (2011):2536-2544. Print.

Chu. *Cargo*. Web. 10 July 2017.

Cox, Kenneth C., Stephen G. Eick, and Taosong He. "3D Geographic Network Displays." *ACM SIGMOD Record* 25.4 (1996): 50-54. Print.

Crampton, Jeremy W. "Interactivity Types in Geographic Visualization." *Cartography and Geographic Information Science* 29.2 (2002): 85-98. Print.

"Designing VR Tools: The Good, the Bad, and the Ugly." *Leap Motion Blog*. 20 May 2015. Web. 10 July 2017. http://blog.leapmotion.com/designing-vr-tools-good-bad-ugly/

Dykes, J.A, and D.M Mountain. "Seeking Structure in Records of Spatio-temporal Behaviour: Visualization Issues, Efforts and Applications." *Computational Statistics & Data Analysis* 43.4 (2003): 581-603. Print.

Elmqvist, N., and P. Tsigas. "A Taxonomy of 3D Occlusion Management for Visualization." *IEEE Transactions on Visualization and Computer Graphics* 14.5 (2008): 1095-1099.

Print.

"Explorations in VR." *Leap Motion Developer*. Web. 10 July 2017. https://developer.leapmotion.com/explorations#110

Hevner, Alan, and Samir Chatterjee. "Design Science Research in Information Systems." *Integrated Series in Information Systems Design Research in Information Systems* (2010): 9-22. Print.

"Home." *MASTERS OF PIE*. Web. 10 July 2017. http://www.mastersofpie.com/

"Is the NASDAQ in Another Bubble? A Virtual Reality Tour of the NASDAQ." *The Wall Street Journal*. Dow Jones & Company. Web. 01 May 2017. http://graphics.wsj.com/3d-nasdaq/

Jenny, Bernhard, Daniel M. Stephen, Ian Muehlenhaus, Brooke E. Marston, Ritesh Sharma, Eugene Zhang, and Helen Jenny. "Design Principles for Origin-destination Flow Maps." *Cartography and Geographic Information Science* (2016): 1-15. Print.

Jenny, Bernhard, Daniel M. Stephen, Ian Muehlenhaus, Brooke E. Marston, Ritesh Sharma, Eugene Zhang, and Helen Jenny. "Force-directed Layout of Origin-destination Flow Maps." *International Journal of Geographical Information Science* (2017): 1-20. Print.

Kinstner, Zach. "[DevUp: Force-Directed Graph VR] Hand-Held User Interface, Audio Feedback, Downloadable Demo." *Medium*. 13 Dec. 2016. Web. 01 May 2017. https://medium.com/@zachkinstner/devup-force-directed-graph-vr-hand-held-user-interface-audio-feedback-downloadable-demo-93bb2dad0bab

Kraak, Menno-Jan. "The space-time cube revisited from a geovisualisation perspective." Proc. 21st International Cartographic Conference. 2003.

Lambert, Antoine, Romain Bourqui, and David Auber. "3D Edge Bundling for Geographical

Data Visualization." *2010 14th International Conference Information Visualisation* (2010). Print.

Médialab Tools. Web. 10 July 2017. http://tools.medialab.sciences-po.fr/seealsology/

Nichols, Sarah, and Harshada Patel. "Health and Safety Implications of Virtual Reality: A Review of Empirical Evidence." *Applied Ergonomics* 33.3 (2002): 251-271. Print.

Phan, Doantam, Ling Xiao, Ron Yeh, P. Hanrahan, and T. Winograd. "Flow Map Layout." *Proceedings of the 2005 IEEE Symposium on Information Visualization (INFOVIS'05)*. Print.

Reda, K., A. Febretti, A. Knoll, J. Aurisano, J. Leigh, A. Johnson, M. E. Papka, and M. Hereld. "Visualizing Large, Heterogeneous Data in Hybrid-Reality Environments." *IEEE Computer Graphics and Applications* 33.4 (2013): 38-48. Print.

Robinson, Arthur H. "The 1837 Maps of Henry Drury Harness." *The Geographical Journal* 121.4 (1955): 440. Print.

"Standardization." *Wikipedia*. Wikimedia Foundation, 08 July 2017. Web. 16 July 2017. https://en.wikipedia.org/wiki/Standardization

Thealphamike. "VR Interface Design Pre-Visualisation Methods." *YouTube*. YouTube, 04 Oct. 2015. Web. 10 July 2017. https://www.youtube.com/watch?v=id86HeV-Vb8

"The Data Visualisation Catalogue." *The Data Visualisation Catalogue*. Web. 10 Apr. 2017. http://www.datavizcatalogue.com/

"The Horrors of Mass Death Are Visualized in This VR Experience." *Creators*. Web. 01 May 2017. https://creators.vice.com/en_us/article/78e5wq/ali-eslami-vr-mass-death-visualization

Thestonefox. "Thestonefox/VRTK." *GitHub*. 14 July 2017. Web. 14 July 2017. https://github.com/thestonefox

Tufte, Edward. “Napoleon’s March”. 10 July 2017. Web. 18 August 2017. https://www.edwardtufte.com/tufte/posters

Virtualitics. "VR Goes to Work." *Virtualitics*. 26 Apr. 2017. Web. 02 May 2017. https://www.virtualitics.com/vr-goes-work/

Ware, Colin. *Information Visualization: Perception for Design*. Waltham, MA: Morgan Kaufmann, 2013. Print.

Ware, Colin, and Peter Mitchell. "Visualizing Graphs in Three Dimensions." *ACM Transactions on Applied Perception* 5.1 (2008): 1-15. Print.

Wood, Jo, Jason Dykes, and Aidan Slingsby. "Visualisation of Origins, Destinations and Flows with OD Maps." *The Cartographic Journal* 47.2 (2010): 117-120. Print.

Zhou, Hong, Panpan Xu, Xiaoru Yuan, and Huamin Qu. "Edge Bundling in Information Visualization." *Tsinghua Science and Technology* 18.2 (2013): 145-156. Print.

Zhu, Xi, and Diansheng Guo. "Mapping Large Spatial Flow Data with Hierarchical Clustering." *Transactions in GIS* 18.3 (2014): 421-435. Print.