Building Immersive Data Visualizations for the Web

Peter W. S. Butcher
University of Chester, UK
Email: p.butcher@chester.ac.uk

Panagiotis D. Ritsos
Bangor University, UK
Email: p.ritsos@bangor.ac.uk

***Abstract*—We present our early work on building prototype applications for Immersive Analytics using emerging standards-based web technologies for VR. For our preliminary investigations we visualize 3D bar charts that attempt to resemble recent physical visualizations built in the visualization community. We explore some of the challenges faced by developers in working with emerging VR tools for the web, and in building effective and informative immersive 3D visualizations.**

I. Introduction

Immersive Analytics (IA) is an emerging research domain that investigates the use of novel display and interface technologies in analytical reasoning and decision making [1]. IA aims to develop collaborative, interactive systems that immerse users in their data, facilitating new ways of making sense of said data, by building upon concepts such as virtual and mixed reality (VR/MR), natural interaction, touch interfaces and context-awareness. VR and MR/AR have, predominately, been focusing on the realistic, graphical portrayal of physical objects, with a few examples attempting to display more abstract information. We believe that, much as VR and MR have been used in the past to display buildings, machines, avatars or physical objects, they can do so with abstract representations of data [2].

Recent research on data physicalization has elaborated on the potential and challenges of physical visualizations [3], bringing abstract data representations to a physical form. Physical representations can be inspected, touched, felt and handed from one person to another, offering users a simple, yet natural interaction style [2]. VR and MR aspire to replicate, and extend, natural and physical interaction through advanced interfaces. When it comes to VR/MR data representation, the user can be immersed in an environment where they can organise data spatially, build 3D data depictions, hold it and inspect it from different angles, much like they would in the physical world. However, unlike physical visualizations, representations in VR can be dynamically altered and manipulated in ways that are not currently possible in the physical world, e.g., filtering, zooming or changing the depicted data without refabrication [2]. Moreover, through being immersed in the data, the sense-making process attempts to harness all of our senses and offer novel ways of understanding our data, compared to traditional 2D data visualization [4].

In this work-in-progress paper, we present our early work on building prototype IA applications, using emerging standards-based web technologies for VR. For this preliminary investigation we have chosen to loosely replicate, in VR, 3D bar charts, such as those from [5]. We specifically focus on web technologies, as we believe these work better in the data visualization ecosystem — which predominately already uses such technologies — compared to game engines often used in VR such as Unity [6] or Unreal [7].

II. Background

As the complexity of data continues to increase, industry and academia are working to find new ways of helping us to understand it. Desktop computers are no longer enough in a world where people are constantly on the move. Our mobile phones, tablets and laptops are curtailing the time we spend using desktops, spawning entirely new research areas such as Personal Visualization and Personal Visual Analytics [8]. Furthermore, research and development of multi-sensory input and output devices are working to push beyond classic *"windows, icons, menus, pointer"* (WIMP) interfaces.

Driven by recent advances in human-computer interfaces, an increasing number of researchers and developers have been creating immersive data visualizations. In 2014, Donalek et al. [9] wrote about "Immersive and collaborative data visualization" highlighting the potential of VR for data analysis. They also discussed iViz, a Unity3D-based data visualizer which attempts to address the lack of optimization for complex data in immersive worlds, such as OpenSim [10]. iViz has full support for contemporary interfaces, such as the Leap Motion [11] and the Oculus Rift [12] headset, and allows multi-user collaborative data exploration. In 2015, Chandler et al. [1] coined the term "Immersive Analytics", discussing usability and design issues of immersive environments for data analytics. Cordeil et al. [13] compare merits and shortfalls of CAVE and head-mounted displays (HMDs) in similar scenarios. Finally, Lu et al. [14] review the current state of mobile immersive environments, relevant to data visualization. Relating concepts, such as beyond-the-desktop, MR visualization have also been discussed [4].

Beyond academic-led researchers, developers and enthusiasts have explored the notion of immersive visualizations. Masters of Pie [15], a UK-based studio, explores big data visualization in VR with vARC, rendering data points onto a spiral, arched over the user. VR Nasdaq [16] uses Three.js and D3.js to present a guided tour of 21 years of the Nasdaq in VR, for Wall Street Journal. Margolis [17] shows how Immersive Analytics can being realized with many different types of software, and using a plethora of display and interface technologies, such as power walls and HMDs, for both VR and MR. He also shows examples of how VR is gradually making its way to the web. Moreover, a growing number of developers in industry are utilising the Web for immersive and interactive data visualization content. Projects such as Adit [18], MathworldVR [19] and CityViewR [20] are all using the emerging WebVR standard, which aims to bring rich VR experiences to the web across all devices and platforms.

III. Platform Considerations

A. The Web as a Development Platform

Increasingly, people need to access their data on all of their devices, at any time, wherever they are. Elmqvist and Irani coined the term "Ubiquitous Analytics" or "Ubilytics" [21], advocating that applications for data analytics should be available from anywhere at any time. As such, applications for Immersive Analytics should be designed to accommodate our growing need for ubiquitous access to our data.

Nowadays, the Web is the most ubiquitous, collaborative and platform-independent way to build and share information [4]. The only requirement for accessing content on the web is a web browser on an Internet-enabled device, removing the need for users to install extra software or applications. As a result, the application and all of its dependencies can be accessed via a URL from anywhere with an Internet connection. All of this makes the web an ideal platform on which to deploy data visualization applications.

B. The Web as a VR Platform

There are lots of powerful data visualization libraries for use in web applications, such as D3.js [22] and Three.js [23]. The latter is currently used in our application, whereas the former is about to be integrated, as we experiment with more datasets. D3.js provides powerful mechanisms for binding and manipulating arbitrary data to a Document Object Model (DOM), rendering visualizations predominately in 2D. Three.js, is a popular framework that uses WebGL, for building VR content for the web, without relying on proprietary browser plugins. Until 2016, building VR content also required a WebVR polyfill [24], providing basic stereo rendering and barrel distortion with a choice of lens configurations for use with smartphones and Google Cardboard [25] viewers. As native support for WebVR becomes more ubiquitous in browsers, the need for the polyfill reduces. Moreover, non-smartphone HMDs are natively supported in 'desktop' browsers that support WebVR [26]. Finally, if a PC or mobile user does not have access to a HMD, WebVR can display a mono-3D scene instead, and that user will still be able to look around in 360 degrees, albeit with reduced functionality.

As WebVR is gaining widespread support with browser vendors, building tools for WebVR content-creation is a hot topic [27]. A number of tools are already available for building rich VR experiences for the web [28], [29], but since the standard is still very young, there is a long way to go before we see libraries, tailored for VR content creation, that are as powerful as those for 2D and pseudo-3D. As the WebVR standard matures, an increasing number of devices and headsets gain support, which will make VR accessible to more people and on more devices than ever.

IV. Design and Implementation

A. Proof of Concept Investigation

Our investigations started long before WebVR was supported in major browsers, and our first prototype application was built using Three.js and the WebVR polyfill [2]. Our focus was to create a proof of concept for low-cost Immersive Analytics using web technologies, with a view of extending to other VR platforms. We built our visualizations to closely resemble their physical counterparts, by including features such as ambient lighting and natural shadows etc. as these have been proven to aid the sense-making process [30]. This prototype renders a split screen, stereo version of a 3D scene when in use with a cardboard viewer (see Figure 1).

The user is positioned in the centre of a virtual world, into which we can load 3D visualizations of different datasets, included in JSON format. As cardboard viewers did not offer interaction input at the time, we rotate the visualization around the vertical axis, for inspection. Filtered data can be depicted by adjusting the dimensions of the base grid and axes, culling the x/y values required. The system employs head tracking with three degrees of freedom (roll, pitch and yaw) for mobile browsers, utilising a devices gyroscope. Desktop users can use mouse controls to view the scene in mono, in 360 degrees.

A major challenge that we faced, when building the prototype, was maintaining a balance between satisfactory frame rates and satisfactory lighting and shadow quality (see Figure 2). Using pure Three.js to render a stereo scene of the same visualization allowed us to maintain 60 frames per second, even with high quality effects. Although the frame rate was satisfactory, image crosstalk and a lack of lens distortion correction led to an uncomfortable viewing experience. These issues are rectified by the introduction of the WebVR polyfill. However, this comes with a trade-off in performance, due to the complexity of additional rendering steps that the browser needs to go through. Nonetheless, whilst using the polyfill affects the frame rate of the application considerably,

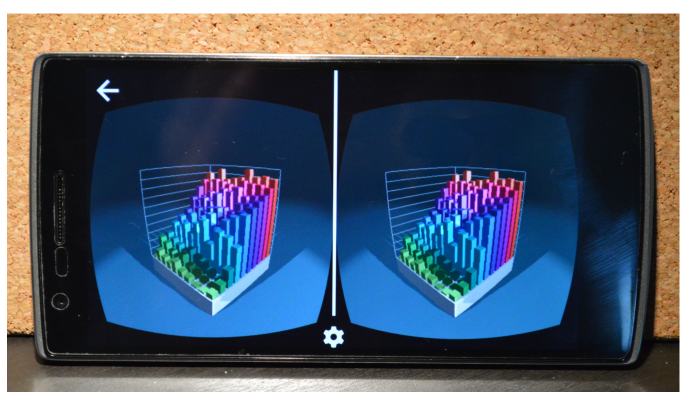

Fig. 1. Example 3D scene from first prototype application rendered with the WebVR polyfill and Three.js, captured inside Google Chrome for Android displaying a 3D bar chart visualization.

when using high quality effects, such as shadows, full-scene antialiasing and smooth lighting, the image clarity is a vast improvement over the pure Three.js implementation. In a heuristic evaluation, the overall experience of the WebVR polyfill version was deemed superior to the plain Three.js version. When using medium to low quality effects we can achieve a constant 60 frames per second on high-end smart phones, for our 1000 polygon scene (500 per eye).

B. Current Ongoing Investigation

Our current investigation is an ongoing continuation from our early experiments with WebVR, which is now natively supported on some desktop and mobile browsers. An increasing number of tools are becoming available for WebVR content creation. For a few months, the most widely adopted framework was A-Frame, an entity-component framework that provides a declarative, extensible, and composable structure to Three.js [28]. As A-Frame uses the DOM as a scene graph, we can apply many of the traditional architectural approaches that have been used in the past to build 3D scenes for the declarative web [31], for example with X3DOM [32]. Moreover, A-Frame provides us with a subset of primitive shapes but also lets us use the full power of Three.js to build our own visualizations. A-Frame lets us write cleaner code, thus allowing us to write VR applications that will scale better, something that is extremely difficult to do with pure Three.js and JavaScript. More recently, Facebook have announced a library for WebVR content creation called ReactVR [29], offering a powerful alternative to A-Frame.

Our ultimate goal is to create interactive and collaborative IA spaces, where many users can interact with 3D data visualizations. In that regard, a large number of user interactions in a single view are required. This has implications in the application's scalability, with particular challenges in state management, edge-cases and validation. Furthermore, making changes to the DOM is computationally expensive, and large applications have to account for numerous concurrent executions, with significant impact on user experience. Since we're working towards rendering potentially large and complex data visualizations at 90Hz, with lots of interactions and visual updates, we needed to find a way to avoid unnecessary re-rendering of the DOM.

One solution to this issue is to build the application using an established framework that offers mechanisms that address these challenges. Our choice is ReactJS [33], a JavaScript library designed for building user interfaces. It provides us with a virtual DOM, which is an abstraction to the HTML DOM, allowing faster updates, when new states are required. This results in simpler, smaller code bases with inherently fewer bugs. React encourages the use of Functional Reactive Programming (FRP) which essentially enables us to write stateless components that react to our application's state. The state usually resides in a single place which simplifies the flow of execution and reduces unexpected errors.

Pairing A-Frame and React together for building our immersive data visualizations has simplified the development process, providing a solid foundation for scaling up our application. Unfortunately, A-Frame does not yet fully support shadows as standard, however we are still able to provide adequate depth cues with simple illumination techniques. Due to an increased rendering performance we were able to add extra complexity to the visualization geometry, such as sharp axis labels, without any impact on performance (see Figures 3 and 4).

Having the ability to interface HMDs to a web browser on a PC now that mainstream browsers are beginning to support the WebVR standard opens up a lot of opportunities for adding rich interactions to our data visualizations. The HTC Vive and Oculus Rift come with handheld controllers which can provide remarkable levels of dexterity. The latest Samsung Gear VR headset and Daydream View [34] ship with a single multi-button hand-held bluetooth controller which will enable more natural interactions on mobile devices. Our application currently utilises the HTC Vive controllers for filtering tasks and inspection. On mobile devices, we utilise a gaze cursor for basic interactions (see Figure 3).

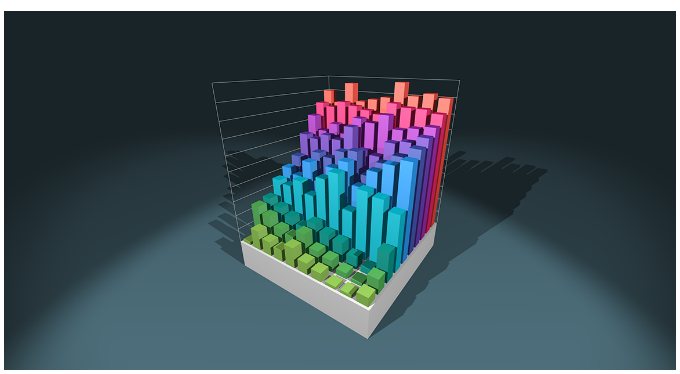

Fig. 2. Example 3D scene from our first prototype application [2] shown in Figure 1. It is rendered with the WebVR polyfill and Three.js, captured inside Google Chrome for desktop displaying a 3D bar chart visualization in mono desktop view with high quality effects. Running the visualization with these settings on a high end smart phone yields unsatisfactory frame rates.

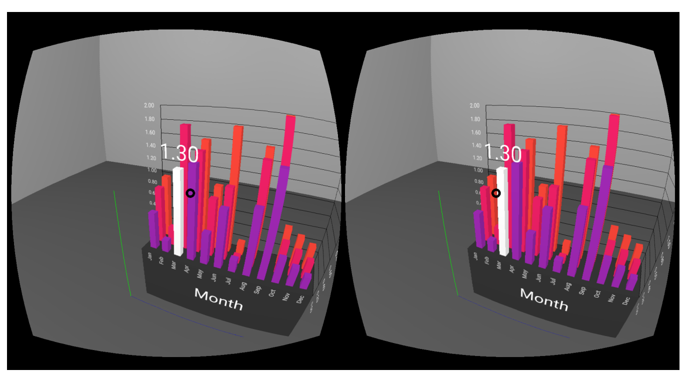

Fig. 3. Example scene from current prototype displaying a 3D bar chart visualization built with A-Frame and React, rendered on Chrome for Android with WebVR enabled. Each bar, when hovered by the gaze cursor, appears in white with a label. The a gaze cursor allows enable simple interactions on mobile and mono desktop setups.

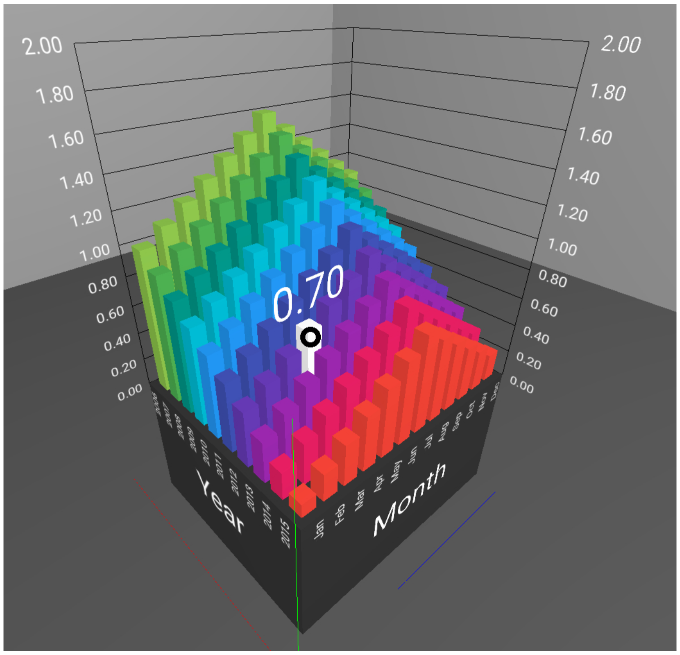

Fig. 4. Example scene from current prototype displaying a 3D bar chart visualization built with A-Frame and React and rendered on Chrome for desktop. The bar hovered by the gaze cursor appears in white with a label.

V. Conclusion and Future Work

Motivated by the recent emergence of technologies enabling WebVR content creation, we explore how they can be used for immersive analytics. In this work, we focus on recreating known physical visualization designs in VR, exploring the performance trade-offs between fast and realistic rendering. The use of WebVR shows promising results, especially as more browsers offer native support for it. As work in the development of WebVR is continuing, and more content creation frameworks become available, we expect similar investigations to increase in number. Our investigation will extend to these frameworks. We also intend to follow up with evaluations of our system with regards to its effectiveness at facilitating visual analytics and sense-making of various different types of datasets and visualizations.

References

[1] T. Chandler, M. Cordeil, T. Czauderna, T. Dwyer, J. Glowacki, C. Goncu, M. Klapperstueck, K. Klein, K. Marriott, F. Schreiber, and E. Wilson, "Immersive analytics," in *Procs. of IEEE BDVA*, 2015, pp. 1–8.

[2] P. W. Butcher, J. C. Roberts, and P. D. Ritsos, "Immersive Analytics with WebVR and Google Cardboard," in *Posters of IEEE VIS*, October 2016.

[3] Y. Jansen, P. Dragicevic, P. Isenberg, J. Alexander, A. Karnik, J. Kildal, S. Subramanian, and K. Hornbæk, "Opportunities and challenges for data physicalization," in *Procs. of ACM CHI*. New York, NY, USA: ACM, 2015, pp. 3227–3236.

[4] J. C. Roberts, P. D. Ritsos, S. K. Badam, D. Brodbeck, J. Kennedy, and N. Elmqvist, "Visualization beyond the desktop–the next big thing," *IEEE CG&A*, vol. 34, no. 6, pp. 26–34, 2014.

[5] Y. Jansen, P. Dragicevic, and J.-D. Fekete, "Evaluating the efficiency of physical visualizations," in *Procs. of ACM CHI*. New York, NY, USA: ACM, 2013, pp. 2593–2602.

[6] Unity Technologies SF. (2017) *Unity3D*. [Online]. Available https://unity3d.com/. Accessed: 18/04/17.

[7] Epic Games Inc. (2017) *Unreal Engine*. [Online]. Available https://unrealengine.com/. Accessed: 18/04/17.

[8] D. Huang, M. Tory, B. A. Aseniero, L. Bartram, S. Bateman, S. Carpendale, A. Tang, and R. Woodbury, "Personal visualization and personal visual analytics," *IEEE TVCG*, vol. 21, no. 3, pp. 420–433, March 2015.

[9] C. Donalek, S. G. Djorgovski, A. Cioc, A. Wang, J. Zhang, E. Lawler, S. Yeh, A. Mahabal, M. Graham, A. Drake *et al.*, "Immersive and collaborative data visualization using virtual reality platforms," in *Procs. of IEEE Big Data 2014*. IEEE, 2014, pp. 609–614.

[10] OpenSimulator. (2017) *OpenSimulator*. [Online]. Available http://opensimulator.org/. Accessed: 18/04/17.

[11] Leap Motion Inc. (2017) *Leap Motion*. [Online]. Available https://www.leapmotion.com. Accessed: 18/04/17.

[12] Oculus VR, LLC. (2017) *Oculus Rift*. [Online]. Available https://www.oculus.com/rift/. Accessed: 13/05/17.

[13] M. Cordeil, T. Dwyer, K. Klein, B. Laha, K. Marriott, and B. H. Thomas, "Immersive collaborative analysis of network connectivity: Cave-style or head-mounted display?" *IEEE TVCG*, vol. 23, no. 1, pp. 441–450, Jan 2017.

[14] A. Lu, J. Huang, S. Zhang, C. Wang, and W. Wang, "Towards Mobile Immersive Analysis: A Study of Applications," in *Procs. of Immersive Analytics Workshop, IEEE VR*, J. Chen, E. G. Marai, K. Mariott, F. Schreiber, and B. H. Thomas, Eds., March 2016.

[15] Masters of Pie. (2017) *Masters of Pie, Winners of the big data VR challenge 2015*. [Online]. Available http://www.mastersofpie.com/big-data-vr-challenge-winners/. Accessed: 13/05/17.

[16] T. Merrel. (2015) *Is the NASDAQ in Another Bubble? A virtual reality tour of the NASDAQ*. WSJ [Online]. Available: http://graphics.wsj.com/3d-nasdaq/. [Accessed: 6-Mar-2017].

[17] T. Margolis. (2017) *How Immersive Analytics Enhance Collaborative Decision-Making*. [Online]. Available https://branch-blog.qlik.com/8c0fa47758a2. Accessed: 18/04/17.

[18] W. Murphy. (2017) *Adit: Enter the Data Mine*. [Online]. Available https://github.com/wmurphyrd/adit. Accessed: 18/04/17.

[19] Sleighdogs Gmbh. (2017) *MathworldVR*. [Online]. Available http://vr.sld.gs/mathworldvr/. Accessed: 18/04/17.

[20] Runhar. (2017) *CityViewR: open data for smart cities, in VR*. [Online]. Available https://github.com/runhar/CityViewR/. Accessed: 18/04/17.

[21] N. Elmqvist and P. Irani, "Ubiquitous analytics: Interacting with big data anywhere, anytime," *Computer*, vol. 46, no. 4, pp. 86–89, 2013.

[22] M. Bostock, V. Ogievetsky, and J. Heer, "D^3 Data-Driven Documents," *IEEE TVCG*, vol. 17, no. 12, pp. 2301–2309, Dec 2011.

[23] R. Cabello. (2017) *Three.js*. [Online]. Available https://github.com/mrdoob/three.js. Accessed: 13/05/17.

[24] Google Inc. (2015) *WebVR Polyfill*. [Online]. Available https://github.com/googlevr/webvr-polyfill/. Accessed: 14/05/16.

[25] Google Inc. (2017) *Google Cardboard*. [Online]. Available https://vr.google.com/cardboard/. Accessed: 13/05/17.

[26] W3C. (2017) *WebVR*. [Online]. Available https://webvr.info/. Accessed: 18/04/17.

[27] M. Hellman. (2015) *Civis Analytics - Exploring Virtual Reality Data Visualization with Gear VR*. [Online]. Available https://civisanalytics.com/blog/data-science/2015/12/04/exploring-virtual-reality-data-visualization-with-the-gear-vr/. Accessed: 29/02/16.

[28] Mozilla. (2017) *A-Frame*. [Online]. Available https://aframe.io. Accessed: 18/04/17.

[29] Facebook Inc. (2017) *ReactVR*. [Online]. Available https://facebook.github.io/react-vr/. Accessed: 18/04/17.

[30] O. H. Kwon, C. Muelder, K. Lee, and K. L. Ma, "A study of layout, rendering, and interaction methods for immersive graph visualization," *IEEE TVCG*, vol. 22, no. 7, pp. 1802–1815, July 2016.

[31] S. Lemme, J. Sutter, C. Schlinkmann, and P. Slusallek, "The basic building blocks of declarative 3d on the web," in *Procs. of Web3D*. New York, NY, USA: ACM, 2016, pp. 17–25.

[32] J. Behr, P. Eschler, Y. Jung, and M. Zöllner, "X3DOM: A DOM-based HTML5/X3D Integration Model," in *Proceedings of the 14th International Conference on 3D Web Technology*, ser. Web3D '09. New York, NY, USA: ACM, 2009, pp. 127–135.

[33] Facebook Inc. (2017) *ReactJS*. [Online]. Available https://facebook.github.io/react/. Accessed: 18/04/17.

[34] Google Inc. (2017) *Daydream View*. [Online]. Available https://madeby.google.com/vr/. Accessed: 18/04/17.