Towards a Framework for Immersive Analytics on the Web

Peter W. S. Butcher*
University of Chester

Nigel W. John†
Univeristy of Chester

Panagiotis D. Ritsos‡
Bangor University

Abstract

We present work-in-progress on the design and implementation of a Web framework for building Immersive Analytics (IA) solutions in Virtual Reality (VR). We outline the design of our prototype framework, <VRIA>, which facilitates the development of VR spaces for IA solutions, which can be accessed via a Web browser. <VRIA> is built on emerging open-standards Web technologies such as WebVR, A-Frame and React, and supports a variety of interaction devices (e.g., smartphones, head-mounted displays etc.). We elaborate on our motivation for focusing on open-standards Web technologies and provide an overview of our framework. We also present two early visualization components. Finally, we outline further extensions and investigations.

Index Terms: H.5.2 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities.

1 Introduction

Immersive Analytics (IA) is an emerging research theme, in which researchers and developers explore visual mechanisms for data exploration and analysis, away from desktop computers and Window-Icon-Mouse-Pointer (WIMP) interfaces [4]. Amongst this multi-flavored research thrust, Virtual Reality (VR) has a prominent place as one of the oldest paradigms for beyond-the-desktop interaction. Despite the fact that VR has been around for decades, the recent emergence of affordable, commercially available head-mounted-displays (HMDs), such as the HTC Vive and Oculus Rift, has reinvigorated the interest in all things VR. Motivated by this evolution of VR, we are interested in building the necessary tools that will enable the visualization community to explore whether, and how, data visualization practices can be enhanced by the use of VR.

In this effort, we present work-in-progress on our IA framework, <VRIA>, which facilitates the development of IA solutions for the web, and is built upon WebVR, A-Frame and React. We are particularly interested in using open-standards Web-based technologies, as we believe that the Web is the most ubiquitous, collaborative and platform-independent way to build and share information [9]. Moreover, such technologies work in synergy with established visualization libraries, through the HTML document object model (DOM). It is no surprise that the most popular technologies for data visualization nowadays (e.g., D3.js [1]) are from the Web technologies ecosystem and are essentially based on open-standards.

2 Background

At the beginning of the millennium Van Dam et al. [10] presented a research agenda that included a call to action on how VR could be an effective medium for scientific visualization. They highlighted the lack of standardization that enables interoperability for the development of VR systems, as a major obstacle. We believe their observation applies to all genres of visualization.

Aided by the recent availability of new affordable VR interfaces, a large number of efforts to employ this paradigm have emerged over the last five years, e.g., [5, 6]. However, very few efforts focus on open-standard web technologies, e.g. [2, 8]. Of particular interest to our work is the Web-based visualization library, Stardust [7], which has a focus on graphical, GPU-driven performance and shares a number of similarities with our work (declarative grammar, web-technologies, WebVR support). However, its main goal is to achieve significant performance improvements for GPU-based rendering of either 2D or 3D content. Our work, however, focuses on providing a flexible and extensible framework for building VR spaces for IA solutions, accessible on the Web, that nonetheless support a variety of interface devices. Another interesting effort is the DXR toolkit (`https://sites.google.com/view/dxr-vis`) which allows prototyping IA visualizations in both AR and VR. DXR shares a similar philosophy to <VRIA>, in terms of prototyping visualizations but is, nonetheless, built around the game engine Unity.

3 The <VRIA> Framework

Following our prior investigations [2, 3], we were motivated to create a framework that would allow anyone to create interactive, immersive WebVR data visualizations. <VRIA> includes methods for rapid prototyping with simple configuration files, as well as an API facilitating the creation of bespoke visualizations and interactions. By accommodating both of these approaches, <VRIA> is suitable for both seasoned programmers and users with little programming experience.

<VRIA> is written in React, a JavaScript library for building user interfaces; Redux, a state container for JavaScript applications; and A-Frame, a framework for building WebVR experiences. The exact structure and implementation of the high-level DOM architecture of an application that makes use of <VRIA> is up to the user, and there is no requirement for the whole application to be written in React. <VRIA> can be integrated into existing applications, with the only requirement being that the overarching application makes use of A-Frame scenes.

The top level node in the example application depicted in Fig. 1 is a React component (🟦) that contains an A-Frame scene component (🟨). The scene component can contain any other A-

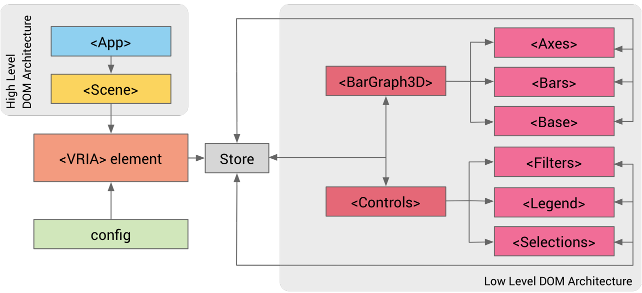

Figure 1: Architectural overview of an example application built with <VRIA>. This diagram also gives a high-level depiction of how data flows through an application.

*e-mail: p.butcher@chester.ac.uk
†e-mail: nigel.john@chester.ac.uk
‡e-mail: p.ritsos@bangor.ac.uk

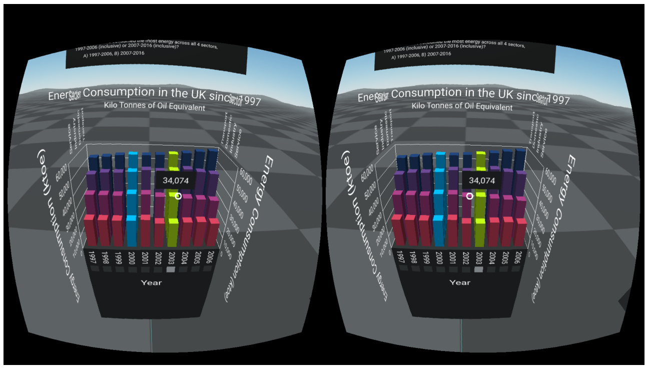
Energy Consumption in the UK since 1997
Kilo Tonnes of Oil Equivalent
34,074
Year
Energy Consumption (ktoe)

Figure 2: A demonstration <VRIA> application running on a mid-range smartphone with Chrome for Android and Google Cardboard. The user is able to fully interact with all controls via a gaze cursor.

Frame components that a user desires. <VRIA> provides a React component called *VRIA* (🟧) which must be placed within the scene element and should be passed a visualization configuration file (🟩). Everything that this component generates is mandated by the configuration file that is passed to it. The store (⬜), is implemented with Redux and contains the application state. It is connected to visualization and control components (🟥). These components map data to React components (🟥), that depict the visualization and all user interfaces with A-Frame (and ThreeJS that underlines it). They also respond to user interaction and report these actions back to the store, which then updates the visualization.

<VRIA> provides a simple method of converting a dataset into WebVR-ready 3D visualizations, coupled with a set of appropriate interfaces (controls), through a declarative format described with JSON. This forms a visualization configuration that our framework can interpret, and allows users to create different visualizations through setting parameters, properties and constraints tailored to each visualization type. In its most basic form, a visualization configuration consists of a dataset and the type of visualization we wish to use. This can either be packaged up as a JSON file or as a JavaScript module. <VRIA> only requires a visualization configuration for basic functionality. However, if extra functionality is required such as a custom set of interactions or a bespoke visualization type, then these can be created with A-Frame and <VRIA>'s API.

4 First Components

<VRIA> currently supports tabular data inputs formatted as JSON or CSV. Data sets may contain any number of fields and records. Supported visualization types currently include 3D bar charts and multivariate scatter plots (Fig. 3) with more abstract data visualization types planned. Each visualization has its own set of possible interactions that can be selectively enabled or disabled in the visualization configuration.

Visualization and control components are written using A-Frame. The framework's API is used to map controls to data transformations, which are then reflected in the visualization. The exact nature of the data transformations and desired interactions are left to the user; the framework then connects everything together. Data transformations for new visualizations can be achieved with existing libraries such as D3.js, or can be written from scratch.

5 Future Work and Conclusions

We present <VRIA>, a framework for building VR spaces for Immersive Analytics, using open-standards Web-technologies. <VRIA> is built using WebVR, A-Frame and React. The resulting VR solutions can be experienced through a WebVR-compliant browser with a variety of devices, ranging from smartphones to HMD-equipped desktop computers. <VRIA> uses a declarative format for specifying visualization types through simple configuration files, simplifying visualization prototyping, data binding and interaction configuration. Our framework is in ongoing development, with a number of enhancements planned. In the next phase we look to build more 3D vis components, along with corresponding interaction mechanisms. We also plan to integrate features that allow collaborative tasks in VR space. More information on <VRIA> is available at `https://github.com/vriajs`.

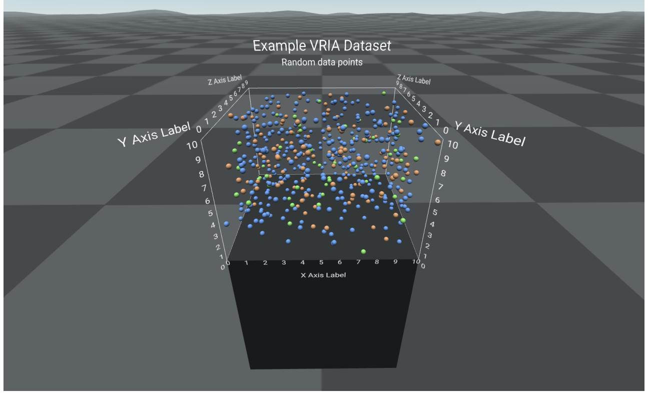

Figure 3: Example of a scatter plot created with <VRIA> depicting a randomly generated multivariate dataset with 500 data points.

References

[1] M. Bostock, V. Ogievetsky, and J. Heer. D^3 Data-Driven Documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, Dec 2011.

[2] P. W. Butcher and P. D. Ritsos. Building Immersive Data Visualizations for the Web. In *Procs. of International Conference on Cyberworlds (CW'17)*, pages 142–145, Sept 2017.

[3] P. W. Butcher, J. C. Roberts, and P. D. Ritsos. Immersive Analytics with WebVR and Google Cardboard. In *Posters of the IEEE Conference on Visualization (IEEE VIS 2016), Baltimore, MD, USA*, 2016.

[4] 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 Big Data Visual Analytics*, (BDVA), pages 1–8, Sept 2015.

[5] M. Cordeil, A. Cunningham, T. Dwyer, B. H. Thomas, and K. Marriott. ImAxes: Immersive Axes As Embodied Affordances for Interactive Multivariate Data Visualisation. In *Procs. of the ACM Symposium on User Interface Software and Technology*, (UIST), pages 71–83, New York, NY, USA, 2017. ACM.

[6] 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 International Conference on Big Data*, pages 609–614, Oct 2014.

[7] D. Ren, B. Lee, and T. Höllerer. Stardust: Accessible and Transparent GPU Support for Information Visualization Rendering. *Computer Graphics Forum*, 36(3):179–188, 2017.

[8] P. D. Ritsos, J. Mearman, J. R. Jackson, and J. C. Roberts. Synthetic Visualizations in Web-based Mixed Reality. In B. Bach, M. Cordeil, T. Dwyer, B. Lee, B. Saket, A. Endert, C. Collins, and S. Carpendale, editors, *Immersive Analytics: Exploring Future Visualization and Interaction Technologies for Data Analytics Workshop, IEEE Conference on Visualization (VIS), Phoenix, Arizona, USA*, Oct. 2017.

[9] 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 Computer Graphics & Applications*, 34(6):26–34, Nov. 2014.

[10] A. van Dam, A. S. Forsberg, D. H. Laidlaw, J. J. LaViola, and R. M. Simpson. Immersive VR for scientific visualization: a progress report. *IEEE Computer Graphics & Applications*, 20(6):26–52, Nov 2000.