

b3.js: A Library for Interactive Web Data Visualizations in Virtual Reality

Huy Le

Ajjen Joshi

Margrit Betke*

Department of Computer Science, Boston University

ABSTRACT

As virtual reality (VR) devices become more ubiquitous, data scientists increasingly recognize the power of viewing and manipulating complex, multivariate data in immersive virtual worlds. In their quest to extract knowledge from the data, data scientists may benefit from visualizations that are easily controlled by hand motion. We here present *b3.js*, a library for creating interactive VR data visualizations on a web browser. The 3D visualizations, for example, 3D scatter, bar, and surface plots, may be viewed through the Oculus Rift and manipulated gesturally with the Leap Motion controller. Oculus Rift provides immersive point of view perspective within the VR environment. Leap Motion enables gestural interaction by allowing users to control the parameters of the virtual camera, for example, modify the point of view. Users can also modify points of a rendered plot. We show that combining the Oculus Rift and Leap Motion with *b3.js* enables users to engage with a complex data set consisting of coherently-organized 3D flight paths of thousands of bats. A novel data exploration experience was created that is difficult to achieve on a flat screen with the traditional mouse-keyboard paradigm.

Keywords: Virtual Reality web 3D graphing library, interactive data visualization, gestural manipulation

Index Terms: H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual Reality;

1 INTRODUCTION

Combined with interactive motion sensor hardware like the Leap Motion controller, VR devices like the Oculus Rift perform together as a new medium and immersive human-computer interface [2, 6]. 3D environments have been created that act as worlds where users not only virtually perceive but also manipulate environmental objects with seamless, intuitive hand motions, for example, in games [3, 4, 8]. Data visualizations in virtual environments for facilitating data science have been considered [5], but the potential of the combination of VR and motion controllers for data science has remained relatively unexplored. The motivation of our *b3.js* project is to expand the usage of VR and motion controllers for data science by providing an interface to visualize and interact with manipulatable datasets in virtual reality. *b3.js* is meant to be incorporated inside web applications to provide VR and motion control capabilities within generated data visualizations. The library enables users to view and manipulate complex real world data, such as the 3D flight paths of thousands of bats (Section 4).

As datasets become larger and more complex, the challenge in data visualization is not only to provide digestible visualization but

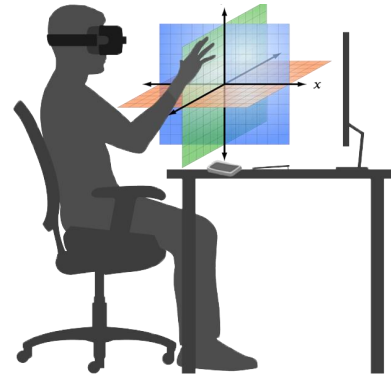


Figure 1: A demonstration of the end user's interaction with a *b3.js*-generated 3D visualization. The user is wearing the Oculus Rift while facing the computer. The Leap Motion controller lies in front of the user where it detects the motion of the user's hand.

also new methods of interacting with these visualizations, in particular in 3D [9]. State-of-the-art 3D visualizations are supported by web graphing libraries that assume that the data will be viewed on a 2D screen. The proposed *b3.js* library enables a designer to create visualizations that can be viewed in a 3D virtual environment.

Users may understand data better if they have controls over the perspective with which they are observing the data. Especially when working with 3D plots, with data points positioned in 3D space, a slight variation of perspective can be deceptive to a user's appropriate understanding of the data. While many 3D scatter and bar plots are represented as raster images that do not enable a user to change the viewing direction of the data, more and more 3D data is represented in a way that specifically enables manipulation of the view point. In particular web browsers can display 3D data in a form that allows change of perspective using mouse controls. The benefits of using hand gestures, over traditional methods, have been explored in the context of many applications [7].

The contributions of our *b3.js* project are twofold: First, the library enables multi-dimensional data visualizations to be rendered on a webpage and immersively viewed using a virtual reality headset. Second, it uses the Leap Motion controller to empower user interactions with the virtual visualization. Users can not only gesturally modify the parameters of the virtual camera to change the point of view, but also naturally interact with the visualized information by selecting and modifying graph elements.

2 B3.JS LIBRARY USAGE

An overview of the *b3.js* library use is given in Fig. 3. Developers who want to work with the *b3.js* library must use a browser capable of detecting VR devices like the Oculus Rift. The two most popular VR-enabled browsers are WebVR-enabled builds of Google Chromium and Mozilla Firefox. After connecting the JavaScript library to HTML, a developer only needs to run *initiate(type)* and *coordinates(name of file)* to create a basic visualization. The currently supported plot types are scatter, bar, and surface plots, and

*Presented at the 23rd IEEE Conference on Virtual Reality, Greenville, SC, March 19–23, 2016. Email: {huyle333, ajjendj, betke}@bu.edu. This work was supported in part by NSF grant 0910908 and 1337866.

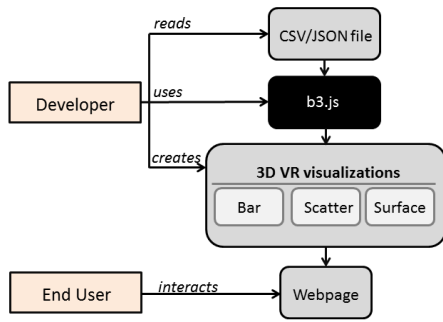


Figure 2: An overview of creating visualizations with *b3.js*.

there is additional support for time series data (Fig. 3). The ingested file to extract the coordinates can be of CSV or JSON formats. Other parameters can be used to incorporate names and additional metadata into the visualization. *b3.js* can be downloaded at <https://github.com/huyle333/b3>.

When data scientists want to work with visualizations created with the *b3.js* library, the Oculus Rift and Leap Motion must be connected to their computers. The Leap Motion lies in front of the computer. The user wears the Oculus Rift and opens an application web page rendering a *b3.js* visualization. The web page detects both the Oculus Rift and Leap Motion. The user can then manipulate the virtual camera’s rotation, zoom, and panning with the Leap Motion controller and see the data in virtual-reality immersion.

3 IMPLEMENTATION AND FUNCTIONALITY

To create three dimensional objects within the web browser, we used a library known as Three.js [1] built on top of WebGL. WebGL is a JavaScript API that provides interactive 2D and 3D graphics renderable inside the web browser without the need of plugins. Using Three.js, we created generating functions for ingesting CSV and JSON datasets that could be rendered into a 3D visualization.

After we finished the design of the first part of our proposed system, we integrated Oculus Rift and Leap Motion controller support. To incorporate the Oculus Rift, we used two libraries called VREffect.js and VRControls.js, one to replicate the screen into two identical parts required to create the 3D effect and the other to translate head coordinates and movement into camera positioning. For the Leap Motion controller, we used leap.js, a javascript library for the Leap Motion controller that detects hand and finger positioning. We created controls based on the number of fingers held outward above the Leap Motion. Using a series of listener functions, we divided functionality into three main components: one finger for selection, two fingers for rotation, and five fingers for zooming in and out. We provided the same camera controls with the mouse.

4 EXAMPLE APPLICATION: VISUALIZING THE COLLECTIVE FLIGHT OF BATS

An example application that we have developed uses real 3D flight paths of a colony of thousands of bats emerging from a cave in the evening. The research objective for analyzing this data was to understand the collective behavior of flying bats. Using immersive VR and motion controls, we wanted to visualize the bat positions and headings in 3D through time. We used *b3.js* to ingest 3D trajectories of flying bats in JSON format and create a 3D-time-series scatter plot. With the visualization created with *b3.js*, we had seamless motion control over the view point so that we could observe the environment and neighboring bats from the perspective of a specific bat. This enabled us to study how neighborhood configurations evolved over time. To follow a particular bat through time, we used the Leap Motion one-finger selection mechanism. Using *b3.js*

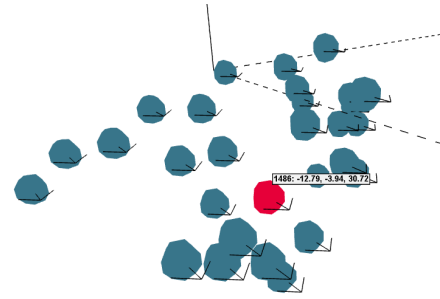


Figure 3: A time-series scatter-plot visualization of the collective flight of a colony of bats. The red sphere denotes a gesturally selected bat. The selection triggers a display of information such as the 3D position and heading of this bat.

to visualize these data provided a level of control and immersion not possible with existing graphing libraries.

5 FUTURE WORK

In this paper, we described our first version of *b3.js*, a library for creating interactive VR data visualizations on a web browser that can be viewed through the Oculus Rift and manipulated with the Leap Motion controller. A future goal for the project is to create a more expansive library with many graphing options and expand support to hardware other than Oculus Rift and Leap Motion. With the current version of *b3.js*, users can create basic 3D scatter, bar, and surface plots. There are options for displaying time series and metadata. *b3.js* has functions to support reading of files in CSV or JSON format. Subsequent versions of *b3.js* will include a more expansive suite of functions, such as changing colors, having full control over the coordinate axes, and creating new data points. Currently, the developer can implement these functionalities by modifying the source or creating custom JavaScript files. We also plan to work with data scientists to further analyze the bat trajectory data using visualizations enabled by *b3.js*.

REFERENCES

- [1] R. Cabello. Three.js. <https://github.com/mrdoob/three.js>, 2010.
- [2] J. C. Coelho and F. J. Verbeek. Pointing task evaluation of Leap Motion controller in 3D virtual environment. In *Creating the Difference, Proceedings of the Chi Sparks 2014 Conference, The Hague, Netherlands*, pages 78–85, 2014.
- [3] A. Espitia-Contreras, P. Sanchez-Caiman, and A. Uribe-Quevedo. Development of a Kinect-based anthropometric measurement application. In *IEEE Virtual Reality 2014, Minneapolis, USA*, pages 71–72, 2014.
- [4] A. Febretti, A. Nishimoto, V. Mateevitsi, L. Renambot, A. Johnson, and J. Leigh. Omegalib: A multi-view application framework for hybrid reality display environments. In *IEEE Virtual Reality 2014, Minneapolis, USA*, pages 9–14, 2014.
- [5] C. Helbig, H.-S. Bauer, K. Rink, V. Wulfmeyer, M. Frank, and O. Kolditz. Concept and workflow for 3d visualization of atmospheric data in a virtual reality environment for analytical approaches. *Environmental Earth Sciences*, 72(10):3767–3780, 2014.
- [6] S. Khattak, B. Cowan, I. Chepurna, and A. Hogue. A real-time reconstructed 3d environment augmented with virtual objects rendered with correct occlusion. In *IEEE Games Media Entertainment (GEM), Toronto, Canada*, pages 1–8, Oct. 2014.
- [7] F. Lamberti, A. Sanna, G. Carlevaris, and C. Demartini. Adding plugable and personalized natural control capabilities to existing applications. *Sensors*, 15(2):2832–2859, 2015.
- [8] M. Vinkler and J. Sochor. Integrating motion tracking sensors to human-computer interaction with respect to specific user needs. In *Central European Seminar on Computer Graphics, cescg.org*, 2014.
- [9] C. Ware and P. Mitchell. Visualizing graphs in three dimensions. *ACM Transactions on Applied Perception*, 5(1):2:1–2:15, 2008.