XRCreator: Interactive Construction of Immersive Data-driven Stories

Donghao Ren
University of California,
Santa Barbara
donghaoren@cs.ucsb.edu

Bongshin Lee
Microsoft Research
bongshin@microsoft.com

Tobias Höllerer
University of California,
Santa Barbara
holl@cs.ucsb.edu

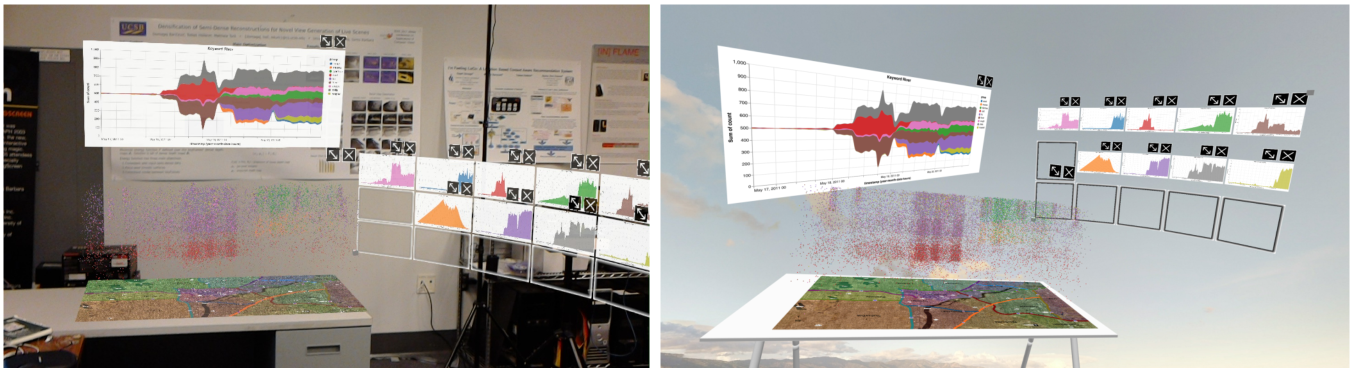

Figure 1: A scene created in XRCreator with the 2011 VAST challenge dataset [2]. Left: the scene viewed in AR (HoloLens); right: the scene viewed in VR (HTC Vive).

ABSTRACT

Immersive data-driven storytelling, which uses interactive immersive visualizations to present insights from data, is a compelling use case for VR and AR environments. We present XRCreator, an authoring system to create immersive data-driven stories. The cross-platform nature of our React-inspired system architecture enables the collaboration among VR, AR, and web users, both in authoring and in experiencing immersive data-driven stories.

CCS CONCEPTS

• **Computing methodologies** → **Mixed / augmented reality**; • **Human-centered computing** → **Visualization toolkits**;

KEYWORDS

Immersive Visualization, Data-driven Storytelling, Visualization Authoring

ACM Reference Format:

Donghao Ren, Bongshin Lee, and Tobias Höllerer. 2018. XRCreator: Interactive Construction of Immersive Data-driven Stories. In *VRST 2018: 24th ACM Symposium on Virtual Reality Software and Technology (VRST '18), November 28-December 1, 2018, Tokyo, Japan.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3281505.3283400

VRST '18, November 28-December 1, 2018, Tokyo, Japan

ACM ISBN 978-1-4503-6086-9/18/11.
https://doi.org/10.1145/3281505.3283400

1 INTRODUCTION

Immersive environments provide many attributes that can be highly beneficial for data-driven storytelling. Presentations can utilize the large space surrounding a standing, sitting, and even walking user, allowing for showcasing a lot of content in an overview fashion and for intrinsic 3D visualizations. The space also enables natural and creative user navigation of these presentations: the user can use natural walking and head movements to find their way in 3D data presentation landscapes.

However, there is currently no tool for authoring immersive data-driven stories except for general-purpose development environments such as Unity, and specific formats such as navigable 360 videos [1]. It is also challenging to develop cross-device applications [7].

We present XRCreator, a prototype system for visualization authors with sufficient knowledge of existing visualization construction techniques to create immersive data-driven stories. XRCreator leverages existing visualization authoring techniques including as Vega, Vega-Lite [6], and Stardust [3] for individual visualization creation, and allows the author to arrange the visualizations in 3D space using a set of layout templates. A simple scene-based state machine engine makes it possible to create presentation sequences, which can be understood as immersive 3D slideshow mechanisms.

XRCreator also supports collaborative authoring, where multiple authors can work using different VR and AR devices, as well as 2D browsers to design presentations that relate to, and can be experienced in, real-world spaces (AR case) or simply utilize to the best extent possible a particular VR tracking space (VR case). The collaborative authoring makes use of the respective strengths of the supported platforms. VR and AR end-users can experience the same presentation together, each relying on different pros and cons of the respective current device capabilities, e.g., small field of view but real-life backdrop, as well as small but high-resolution graphics for the AR user versus large field of view, but low-resolution graphics, as well as a virtual backdrop scene for the VR user [4].

2 DESIGN

A conceptual framework is essential for creating an authoring system. XRCreator's framework is inspired by Ellipsis [5], which is an influential authoring system for creating narrative visualizations using traditional WIMP-style user interfaces. In Ellipsis, a story is a set of components (visualization, control widgets, and annotations) coordinated by a state machine. As a first step towards authoring immersive data-driven stories, we use a simplified version of this framework and add components specific to immersive storytelling.

In XRCreator's framework, a *story* consists of a set of *scenes*. Each scene consists of a set of *components*, including *visualizations*, *layouts*, and *transition widgets*. A *visualization* can be a 2D chart laid out in 3D or a true 3D chart; A *layout* places visualizations in uniform ways. For example, a grid layout places visualizations in an $N \times M$ grid. A *transition widget* can execute scene transitions, such as going to the next scene.

3 IMPLEMENTATION

XRCreator's software architecture enables collaborative authoring of immersive data-driven stories using multiple devices. It consists of an *application server* that maintains the story, performs the application logic, synchronizes the displayed content with the AR/VR/Desktop devices, and receives user input from the VR controllers, devices, as well as commands from the desktop user interface (Figure 2).

We built a component framework mimicking React to manage content for the immersive space. Consider this a custom version of the React library that produces a custom *scenegraph* instead of DOM elements. This approach is similar to React 360 (https://facebook.github.io/react-360), except that we separate the application logic and rendering to different machines and synchronize the generated scenegraph. We also use a Flux architecture that consists of a store, a set of components, and a dispatcher to implement the application logic. The store maintains the designed story and the states of user interface elements. The components display the different parts of the story and enable a 3D user interface for selection and navigation by coordinating data with the store. Upon user interaction, the components send actions through the dispatcher to the store, which updates its states and then notifies the components to update themselves. Once components are updated, the scenegraph is produced and synchronized across the connected devices. Our implementation enables a unified coding paradigm throughout the system, making the implementation easy and less error-prone.

Since the HoloLens and the HTC Vive use different coordinate systems, we implemented a basic calibration mechanism to align them together so that they share the same immersive space. Calibration is completed by aligning a physical Vive controller to a 3D model of it shown in the HoloLens. After repeating this six times, the system figures out a transformation matrix that bridges the two

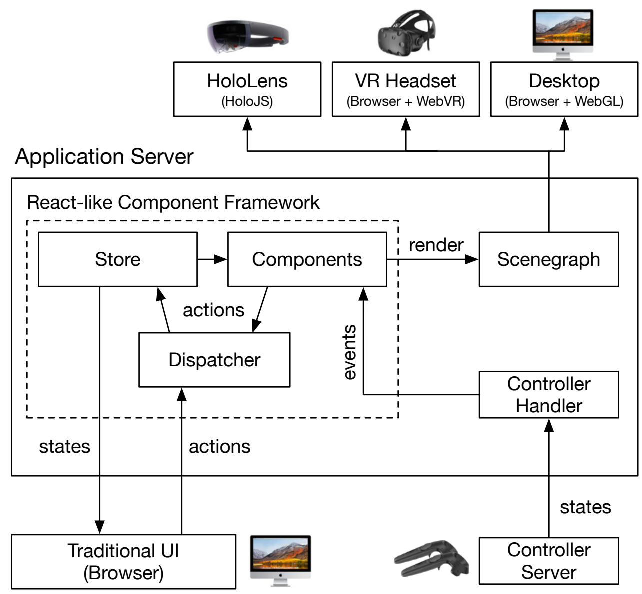

Figure 2: The architecture of XRCreator.

coordinate systems. Once calibration is completed, we continuously update it by using the HoloLens' SpatialAnchor API.

4 CONCLUSION

In this extended abstract, we have discussed the design and implementation of XRCreator. In the future, we would like to further expand its capabilities and conduct evaluations.

ACKNOWLEDGMENTS

This work is supported by the US National Science Foundation under Grant IIS-1748392, and in part by ONR grant N00014-16-1-3002.

REFERENCES

[1] Jacqueline Chu, Chris Bryan, Min Shih, Leonardo Ferrer, and Kwan-Liu Ma. 2017. Navigable Videos for Presenting Scientific Data on Affordable Head-Mounted Displays. In *Proceedings of the 8th ACM on Multimedia Systems Conference (MMSys'17)*. ACM, New York, NY, USA, 250–260.

[2] Georges Grinstein, Kristin Cook, Paul Havig, Kristen Liggett, Bohdan Nebesh, Mark Whiting, Kirsten Whitley, and Shawn Knoecni. 2011. VAST 2011 Challenge: Cyber security and epidemic. *IEEE VAST* 2011 (2011), 299–301. http://www.cs.umd.edu/hcil/varepository/benchmarks.php#VAST2011, accessed August 2018.

[3] Donghao Ren, Bongshin Lee, and Tobias Höllerer. 2017. Stardust: Accessible and Transparent GPU Support for Information Visualization Rendering. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 179–188.

[4] Joan Sol Roo and Martin Hachet. 2017. One Reality: Augmenting How the Physical World is Experienced by Combining Multiple Mixed Reality Modalities. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17)*. ACM, New York, NY, USA, 787–795.

[5] Arvind Satyanarayan and Jeffrey Heer. 2014. Authoring narrative visualizations with ellipsis. In *Computer Graphics Forum*, Vol. 33. Wiley Online Library, 361–370.

[6] Arvind Satyanarayan, Dominik Moritz, Kanit Wongsuphasawat, and Jeffrey Heer. 2017. Vega-lite: A grammar of interactive graphics. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (2017), 341–350.

[7] Maximilian Speicher, Brian D. Hall, Ao Yu, Bowen Zhang, Haihua Zhang, Janet Nebeling, and Michael Nebeling. 2018. XD-AR: Challenges and Opportunities in Cross-Device Augmented Reality Application Development. *Proc. ACM Hum.-Comput. Interact.* 2, EICS, Article 7 (June 2018), 24 pages.