Characterizing Bottlenecks towards a Hybrid Integration of Holographic, Mobile, and Screen-based Data Visualization

Alexander Shearer[*] Lei Guo[†] Junshu Liu[‡] Megumi Satkowski[§] Robert LiKamWa[¶]

School of Arts, Media & Engineering
Arizona State University
Tempe, AZ

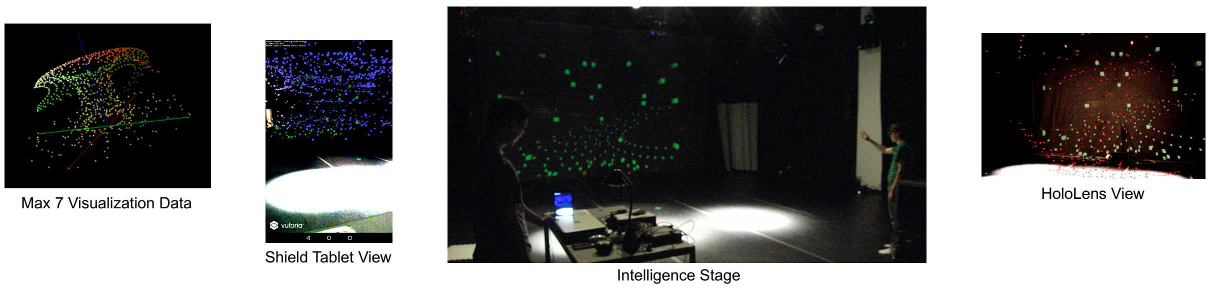


Figure 1: Data visualized across NVIDIA Shield Tablet, Intelligence Stage at Arizona State University, and Microsoft HoloLens

ABSTRACT

The rapid advance of handheld and head-mounted mixed reality technologies comes with many opportunities to integrate 3D visualizations on top of screen-based 2D technologies for immersive analytics, such as the ability to walk through particle data. However, while mixed reality devices can render virtual worlds into the three dimensional space of the physical world, integrating these devices with 2D immersive technologies is fundamentally constrained by the limited computing and networking resources of energy-efficient mobile devices. Towards a better understanding of these limitations, we design and implement a distributed particle rendering framework to bridge mobile mixed-reality devices with an immersive screen-based stage environment using a Microsoft HoloLens, an NVIDIA Shield Tablet, and a multi-projector immersive data visualization environment. Using our framework, we characterize rendering and networking constraints of the devices involved in the hybrid data visualization. We propose strategies to operate within these constraints to provide rich immersive 3D visualizations through dynamic partitioning and selective rendering of data-visualization workloads across 2D and 3D devices.

Keywords: Augmented Reality, Mixed Reality, Mobile Systems, Hybrid 2D/3D Data Visualization.

1 INTRODUCTION

Due to advances in mobile depth sensors and head-mounted projection optics, several entities have released commercial mixed reality devices, including the Microsoft HoloLens [8] and Meta [7] headsets. Similarly, camera-based computer vision technologies have enabled a proliferation of smartphone- and tablet-based augmented reality frameworks, including PTC Vuforia [9] and Apple ARKit [1]. These platforms introduce many opportunities to richly integrate 3D virtual objects among 2D screen-based displays, e.g., CAVE automatic virtual environments [3] and the WILD multisurface interactive environment [2]. As shown in Figure 1, we design a framework to immerse a user in a dense field of particles in a hybrid 2D/3D environment. However, such environments are fundamentally constrained by: (i) the limited computing resources of mobile devices, and (ii) the networking latency and throughput of distributing data.

In this paper, we describe our prototype integration framework testbed (§2), characterize computation and networking bottlenecks (§3), and propose potential solutions to overcome limitations through selective rendering and dynamic partitioning (§4).

We have released source code, video demonstrations, and implementation details at `http://meteor-studio.com/holostage`.

2 PROTOTYPE FRAMEWORK

We build a framework to investigate the hybrid integration of head-mounted and handheld mixed-reality devices with immersive screen-based environments. Our platform integrates three devices, which we enumerated here:

- A *Microsoft HoloLens* head-mounted mixed-reality device, using depth cameras and other sensors to position virtual objects in a real environment. The HoloLens employs an Intel Atom processor and a specialized holographic processing unit to render visualizations.
- A *NVIDIA Shield Tablet K1*, running Android 7.0 on a Tegra K1 System-on-Chip. The Shield Tablet features powerful graphics performance on top of a mobile sensor package, including an inertial motion unit, a front-facing camera, and a rear-facing camera.
- The *iStage (Intelligence Stage)*[1] at Arizona State University, consisting of motion capture, controllable lighting, and immersive projection covering a 10m x 10m floor and a 10m x 8m screen. Data management and projection runs on Max 7 software on Mac Pro computers.

[*]e-mail: acshear1@asu.edu
[†]e-mail: lei.guo.4@asu.edu
[‡]e-mail: jliu237@asu.edu
[§]e-mail: asatkows@asu.edu
[¶]e-mail: likamwa@asu.edu

[1]https://artsmediaengineering.asu.edu/about/facilities/intelligence-stage-istage

To distribute visualization data, we implement a networking protocol to send and receive frames of particles, encoding x,y,z positions, scales, rotations, and RGB colors. We develop our protocol implementations to be compatible with Max 7, HoloLens, and Android. Our devices send TCP transmissions on a Wi-Fi network.

We develop software to render particle visualizations for each device, using the Unity Game Engine for the HoloLens and C++ bindings of the Vulkan Graphics API for the NVIDIA Shield Tablet. Our software uses the PTC Vuforia SDK to geometrically register the devices, creating a uniform coordinate system for the virtual environments. In addition to providing the ability to visualize scientific and creative data, our multi-device platform serves as a testbed to explore system limitations.

3 Systems Bottlenecks and Opportunities for 2D/3D Hybrid Visualization

Providing data visualization across multiple mobile devices introduces fundamental bottlenecks, especially in render workload computation and data distribution. Towards designing a framework for effective hybrid visualization, we characterize these bottlenecks on modern commercially-available devices.

3.1 Rendering Visualization Workloads

The ability to create and display complex data is tied to the computational ability of the mobile device. Mobile system-on-chips for smartphones, tablets, and head-mounted devices typically incorporate a GPU for accelerated graphics, efficiently rendering the state of the virtual environment to the frame buffer of the display device. However, while mobile graphics pipelines have grown to support ever-increasing graphics workloads, large object manipulation workloads are also a computational burden, especially due to signficant memory transactions. This forces a reduction in rendering frame rate to maintain a steady pace of display updates.

In our HoloLens characterization, charted in Figure 2a, we find that without networking, the device can randomly update and render a frame of up to 1000 particles at 33.3 milliseconds per frame, attaining a frame rate of 30 frames per second (fps). Meanwhile, randomly placing and rendering $17,500$ particles uses 1.08 seconds per frame, dropping the frame rate to below 1 fps.

Similarly, as charted in Figure 2b, the Shield Tablet can randomly generate and render up to 2500 particles at 31.0 milliseconds per frame, attaining 30 fps. Rendering $25,000$ particles requires 155 milliseconds per frame, dropping the frame rate to around 6.5 fps. These rates are dependent on particles being visible in the viewport of the virtual camera in the environment; objects outside of the viewport do not significantly affect rendering speed.

3.2 Distributing Visualization Data

Data transmission between multiple devices is a notorious impediment to performance and energy-efficiency. Device-to-device connectivity is straightforward to establish; we leverage standard TCP protocols over Wi-Fi. However, as visual data carries a large footprint, continuously distributing dense visualizations over a network implies system requirements of high throughput and low latency.

For our characterizations, we network our devices through a 450 Mbps wireless router. Our server sends frames of particles, using 12 bytes per particle to encode position coordinates along with a 300 byte header. Including packet processing time to store position coordinates, our HoloLens implementation uses 9.02ms per particle to process a received frame, while the Shield Tablet implementation uses 0.77ms per particle to process a received frame.

Notably, as charted in Figures 2a and 2b, our measurements indicate that while networking overhead is not negligible, both the HoloLens and the Shield Tablet are significantly bottlenecked by rendering overhead. That is, for large numbers of particles, the devices can receive more data than they can render.

4 Proposed Research

The characterized measurements motivate a need to design mechanisms to improve the perceived spatiotemporal density of the visualization, despite computational and networking constraints. We propose an investigation of various framework mechanisms to be implemented across both the server and mobile devices.

4.1 Selective Rendering Density

Spatiotemporally dense representations require significant system resources to render, as characterized in Section 3. However, a low density of particles may be sufficient to represent many regions of the data, whereas a high density may be needed for other groups of particles. Driven by such measures as perceptual sensitivity to spatial gradients and variation of movement of the data [5, 6, 12], a selectively dense rendering decision on the mobile device would allow a reduction in rendering workload, allowing raised system performance. Specific to mixed-reality visualization, the density of rendered particles will need to account for the positioning of the camera viewport, considering particles that are visible.

For effective visualization on resource-constrained devices, we plan to study the challenges of: (*i*) dynamically prioritizing density for regions of particles based on a mixed-reality user's position and orientation, (*ii*) ensuring visually smooth transitions in space and time, and (*iii*) evaluating perceptual effects of the modified visual representations.

4.2 Partitioning 2D/3D Visualization Workloads

Our framework creates the opportunity for hybrid 2D and 3D visualizations, which introduces questions of which content to display in the different modalities, i.e., 2D/3D partitioning. While load balancing for multi-device rendering performance has been well-explored [4, 10, 11, 13], the visualization partitioning for mixed-reality devices also affects the user's visual experience. Particles far beyond the plane of the 2D screens are less sensitive to perspective inaccuracies, and can thus be projected on the surrounding 2D screens. Conversely, particles close to the user should leverage the mixed-reality device for rapid fine adjustments to user positioning.

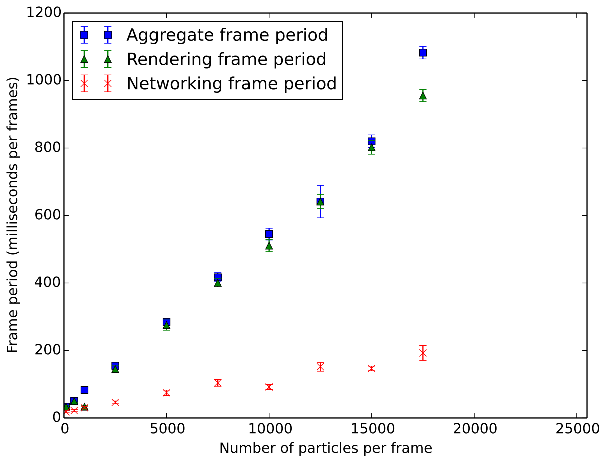
Thus, towards a partitioning mechanism for hybrid screen-based/mixed-reality immersive visualization, we plan to study: (*i*) the effects of mixed-reality user movement and geometric registration latency on particle placement accuracy, (*ii*) the implications of partitioning on rendering performance, (*iii*) the influence of partitioning on selective density decisions (Section 4.1), and (*iv*) partitioning 2D/3D for multiple mixed-reality users.

5 Conclusion

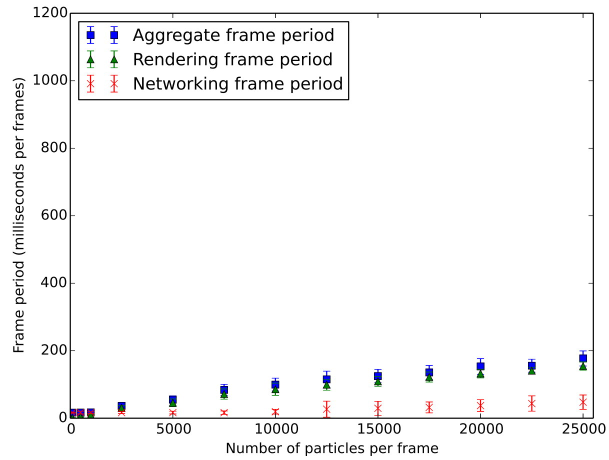
By characterizing our hybrid 2D/3D particle framework, we have confirmed that visualization is limited by rendering and networking constraints. On our implementations, the computational burden of rendering poses a performance bottleneck for both the NVIDIA Shield tablet and the Microsoft HoloLens. Built on these observations, we propose to study selective density and data partitioning strategies to reduce system overhead while improving visual experience. Through investigating these and other strategies, we hope to deploy an adaptive framework with optimization mechanisms be implemented on server computers and mobile devices. The resulting distributed rendering system will generate rich hybrid 2D/3D visualizations that can be readily used for particle simulations, artistic experiences, and other scientific and creative visualizations.

Acknowledgments

The authors thank Connor Rawls, Brandon Mechtley, and Sha Xin Wei at the Synthesis Center at Arizona State University for their use of the Intelligence Stage and scientific dataset. The authors also thank NVIDIA for their support in providing devices to our lab.



(a) Visualization frame period on Microsoft HoloLens



(b) Visualization frame period on NVIDIA Shield

Figure 2: Mean and standard deviation of visualization frame period (inverse of throughput) on mobile mixed-reality devices over 100 frame samples. Rendering frame period is the time between rendering sequential frames of randomly generated particles, including random number generator latency. Networking frame period is the time between receiving sequential particle frames, including the processing latency of updating computational state. Aggregate frame period is the time between the rendering of sequential particle frames received from the server, with rendering and networking active in separate threads.

REFERENCES

[1] Apple. https://developer.apple.com/arkit/.

[2] M. Beaudouin-Lafon, S. Huot, M. Nancel, W. Mackay, E. Pietriga, R. Primet, J. Wagner, O. Chapuis, C. Pillias, J. Eagan, et al. Multisurface interaction in the wild room. *Computer*, 45(4):48–56, 2012.

[3] C. Cruz-Neira, D. J. Sandin, T. A. DeFanti, R. V. Kenyon, and J. C. Hart. The cave: audio visual experience automatic virtual environment. *Communications of the ACM*, 35(6):64–73, 1992.

[4] G. Humphreys, I. Buck, M. Eldridge, and P. Hanrahan. Distributed rendering for scalable displays. In *Proceedings of the 2000 ACM/IEEE conference on Supercomputing*, p. 30. IEEE Computer Society, 2000.

[5] M. S. Langer, J. Pereira, and D. Rekhi. Perceptual limits on 2d motion-field visualization. *ACM Trans. Appl. Percept.*, 3(3):179–193, July 2006. doi: 10.1145/1166087.1166090

[6] T. McLoughlin, R. S. Laramee, R. Peikert, F. H. Post, and M. Chen. Over two decades of integration-based, geometric flow visualization. In *Computer Graphics Forum*, vol. 29, pp. 1807–1829. Wiley Online Library, 2010.

[7] Metavision. https://www.metavision.com.

[8] Microsoft. https://www.microsoft.com/en-us/hololens.

[9] PTC. https://www.vuforia.com.

[10] R. Samanta, T. Funkhouser, K. Li, and J. P. Singh. Hybrid sort-first and sort-last parallel rendering with a cluster of pcs. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware*, pp. 97–108. ACM, 2000.

[11] R. Samanta, J. Zheng, T. Funkhouser, K. Li, and J. P. Singh. Load balancing for multi-projector rendering systems. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware*, pp. 107–116. ACM, 1999.

[12] B. Spencer, R. S. Laramee, G. Chen, and E. Zhang. Evenly spaced streamlines for surfaces: An image-based approach. In *Computer Graphics Forum*, vol. 28, pp. 1618–1631. Wiley Online Library, 2009.

[13] Y. Wang, X. Sang, K. Wang, B. Yan, and C. Yu. Synchronized rendering of super-multiview videos for the frontal projection three-dimensional display. In *Digital Holography and Three-Dimensional Imaging*, pp. DW4A–4. Optical Society of America, 2015.