Massive Data Visualization Techniques for use in Virtual Reality Devices

Jason A. Ortiz*† – Joseph A. Insley*¶ – Janet Knowles* – Victor A. Mateevitsi* – Michael E. Papka*¶ – Silvio Rizzi*
*Argonne National Laboratory †University of Central Florida ¶Northern Illinois University

Introduction

- Visualizing scientific simulations executed on supercomputers producing massive amounts of data is essential to discovery and dissemination.
- Methods for transforming and displaying such large data and its visualizations for use in Extended Reality (XR) devices are not commonly supported.
- We have created a proof-of-concept Virtual Reality (VR) application with Unity using Universal Scene Description (USD) files exported from SideFX Houdini to display and interact with large time-varying scientific data visualizations in VR.

Methods

- We surveyed three XR applications capable of displaying scientific data visualizations: KitWare ParaView VR (Figure 1), SciVista SummitVR, and Nvidia Omniverse XR.
- Experimenting with these tools led to further exploration of USD as a file format for scientific visualization data because it efficiently stores meshes, textures, shading, animations, etc.
- Using Unity's XR Interaction Toolkit and USD packages allow for a cohesive and simple VR experience to play or pause USD `timeSamples`.
- Figure 2 shows the workflow used.

Workflow

1. Execute HPC MPI simulation and store .npy binary files.
2. Import .npy files directly into Houdini, then create visualizations of timeseries data.
3. Export *N* .usd binaries via USD render node per timestep.
4. Use `usdstitch` script to combine *M* different USD files into one new USD file *K* times where $K = \frac{N}{M}$.
5. Import *K* USDs via Unity USD package as prefabs and add colliders to the meshes.
6. Create a `Timeline` asset that plays *K* USD prefab assets.
7. Add XR Ray Interaction callbacks for custom script to play or pause `PlayableDirector` on `Select`
8. Play Unity application on compatible VR device.

Results

- Two .npy data files are ~108 MB each. When exported from Houdini to USD binary, each file is ~18 GB. After import to Unity, each stitched USD is translated into a ~89 MB prefab asset that has an external reference to the original ~18 GB stitched USD.
- When the application is paused, frames per second (FPS) is ~70, but when playing, FPS is ~7. FPS drop might be due to CPU-time accessing USD data.
- Apparent motion-to-photon latency seems negligible but requires further investigation.
- Figure 3 shows the end-user experience successfully playing and pausing scientific visualization animations.
- VR application serves as a proof-of-concept for a workflow to display and interact with time-varying scientific data visualizations.

Major Questions

- What combinations of software and file formats produce a viable workflow to display and interact with massive scientific data visualizations in XR devices?
  - Are there file size constraints?
  - What type of data can be displayed?
- How can XR applications enable novel scientific discovery, dissemination, and collaboration?
- What is the state of the art for scientific data visualization in XR?

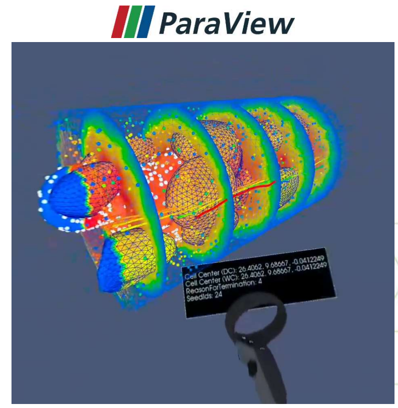

Figure 1: Interacting with a blood flow visualization in ParaView VR.

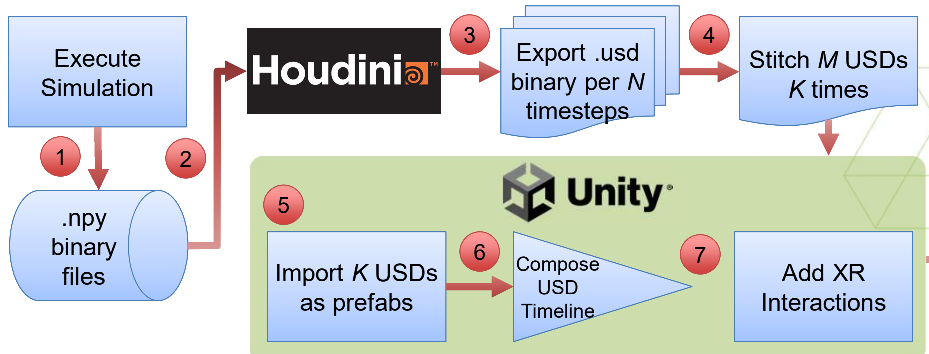

Figure 2: Workflow to create scientific data visualization XR app.

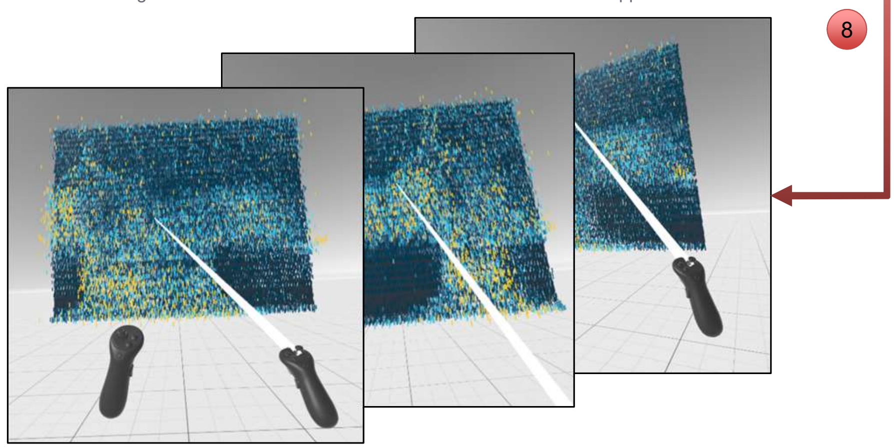

Figure 3: VR application running on Meta Quest 2 tethered by Quest Link USB-C cable to PC with Nvidia RTX 3080 GPU showing three frames of a user playing or pausing an animation of activated neurons of a neural network training on LIGO gravitational wave datasets.

Future Work

- Improve FPS through level-of-detail and/or mesh decimation before import.
- Add rotation, translations, and scaling of the assets in Unity using XR interactions, as well as animation time reversal and scrubbing.
- Investigate loading USD prefabs from remote sources using Unity Addressables package to enable remote app playing.
- Improve graphics quality via the Unity Universal Render Pipeline and/or High Definition Render Pipeline.
- Try launching application on MR and AR devices, such as the Microsoft HoloLens, and gathering performance metrics.

Related Work

[1] M. A. Bolstad, "Large-Scale Cinematic Visualization Using Universal Scene Description," in *IEEE 9th Symposium on Large Data Analysis and Visualization (LDAV)*, Vancouver, BC, Canada, Oct. 2019, pp. 1–2. doi: 10.1109/LDAV48142.2019.8944362.

Data courtesy of Eliu Huerta, Pranshu Chaturvedi, Asad Khan, Nikil Ravi, and Minyang Tian with Data Science and Learning at Argonne National Laboratory. This work was supported by and used resources of the Argonne Leadership Computing Facility, which is a U.S. Department of Energy Office of Science User Facility supported under Contract DE-AC02-06CH11357.