



Virtual Reality in Unity as a Tool for Scientific **Computing and Visualization**

Brandon Horton, Erick Moen, Michael Qian, Aiichiro Nakano

Introduction

- The Importance of Visual Data: • Anscombe's Quartet
- $2D \rightarrow 3D$ lacksquareNatural venue for VR

 \bullet

- The CAVE: \rightarrow Bulky and Expensive (Obsolete?)
- VR Head-Mounted Device \rightarrow Inexpensive and Portable



Tools

- Modern Game Engine: ulletUnity or Unreal
- VR HMD: \bullet Oculus, HTC Vive
- Controller: \bullet Leap Motion, Xbox

Workflow for

Scientific VR in Unity

Generate Objects



Immersive Data Visualization

- VMD-generated object files quickly loaded into Unity
- Leap Motion replaces keyboard and mouse for more direct controller experience
- Unity vertex and fragment shaders help dynamically render pieces of the desalination membrane
- Rendering plane follows head movement via Oculus for intuitive viewing control





Beyond Visualization: Active Simulation

- Not limited to visualization \rightarrow Unity capable of executing simulations
- C# Kinetic Monte Carlo Code spawned on new thread
- Simulation populates an animation queue, while Unity pops according to user-specified rate • → Interactive Simulation Steering
- Result: Simultaneous simulation and rendering in real-time



Future Directions

- Direct pipelining from VMD to Unity from the command line \rightarrow Direct parsing of simulation data in Unity
- Generic Kinetic Monte Carlo Visualization
 - \rightarrow Beyond biological electron transfer
- Visualization of High Performance Simulations via Server-Client Communication with Unity



Special Thanks to Masato Nakano, Moh El-Naggar, Hye Suk Byun, and Tao Wei

Ming Hsieh Institute Ming Hsieh Department of Electrical Engineering

Collaboratory for Advanced Computing and Simulations

CACS