ROOT VR: Immersive Data Analysis and Interaction

Professor Jay Hauser, Dr. Muhammad Ansar Iqbal,
Aryan M. Pour

UCLA Department of Physics and Astronomy

November 9, 2025

UCLA

Introduction

- **UCLA Physics VR**: Utilize VR for (particle) physics
- Existing main projects: CMS VR (CMS detector), G4VR (GEANT 4), VR Spy (iSpy)

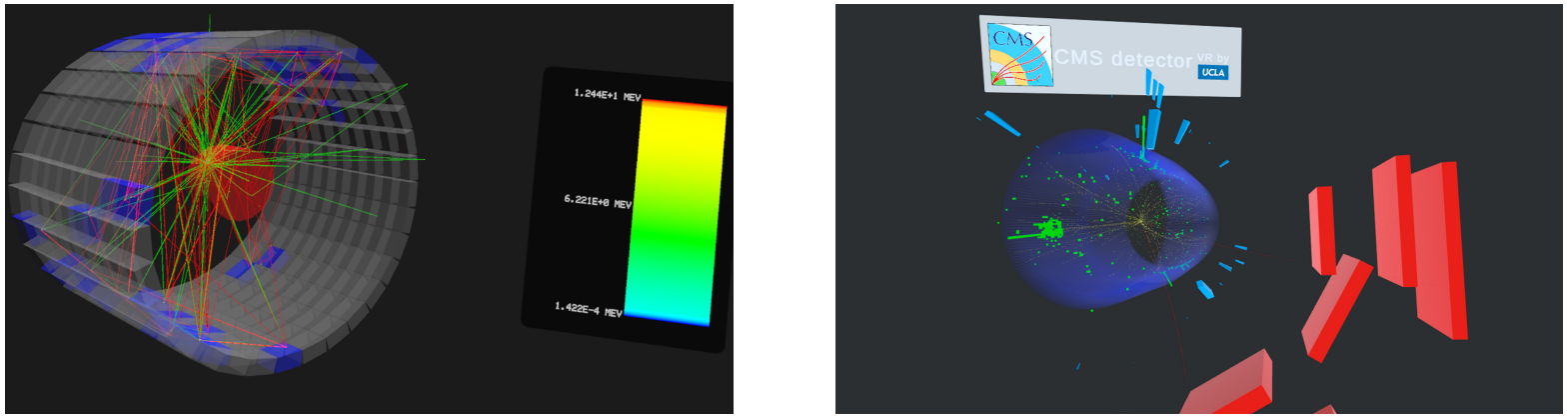

Figure 1: Existing UCLA Physics VR projects

Goals of Data Analysis Software

- **Play a role in the understanding of relationships between variables**

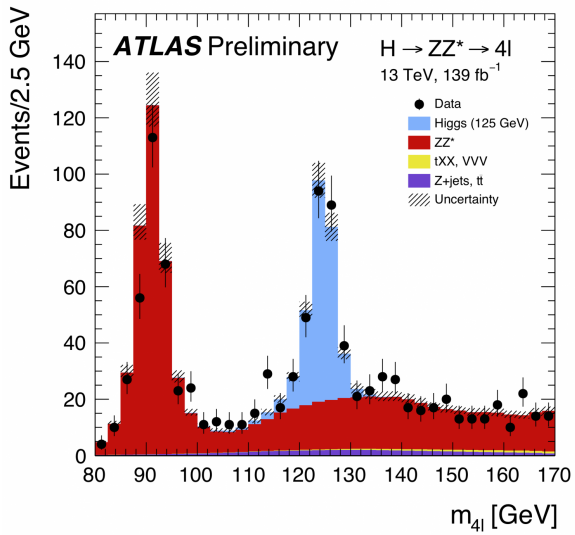

Figure 2: Example relationship – 4 lepton mass vs event frequency

Goals of Data Analysis Software

- **Play a role in the understanding of relationships between variables**
- Fit data to make predictions

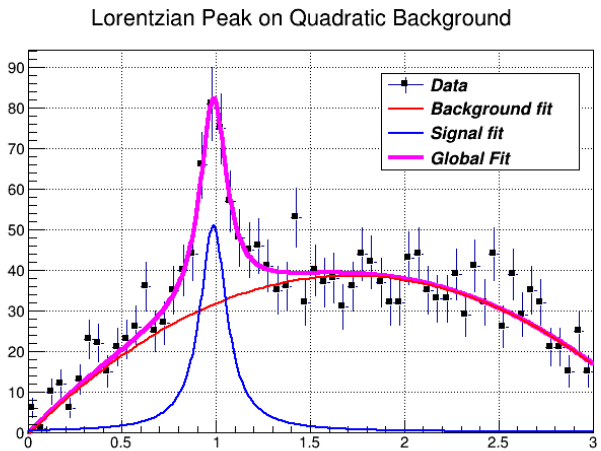

Figure 3: Example fit

Goals of Data Analysis Software

- **Play a role in the understanding of relationships between variables**
- Fit data to make predictions
- Interact with data from command line: add data, remove bins, etc.

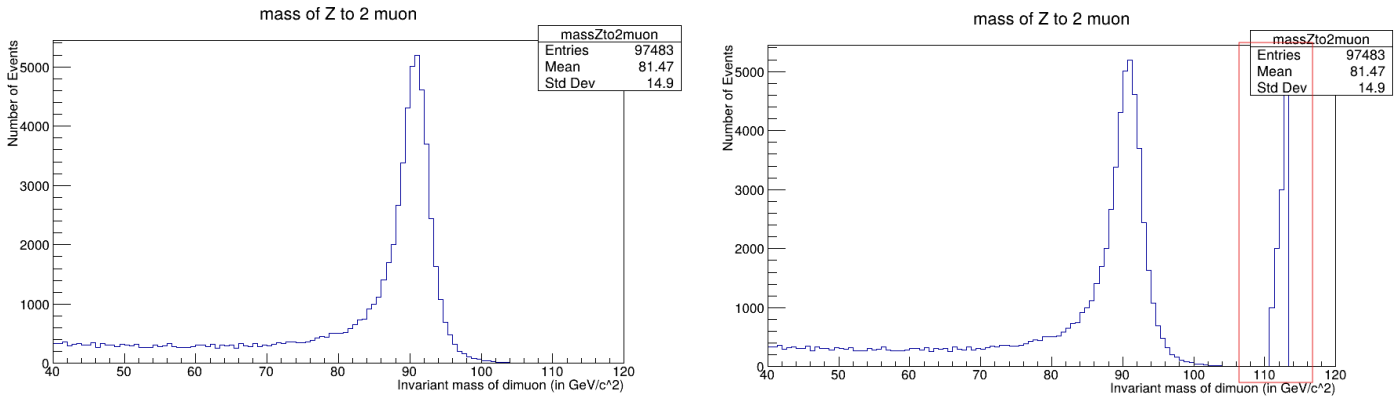

Figure 4: Adding (arbitrary) data entries

UCLA

VR Data Analysis

- Spatial immersion and perspective
  - Inherently more intuitive understanding of relationships between variables
  - Natural evolution: raw numbers $\rightarrow$ Screen visuals (seeing data) $\rightarrow$ World visuals (being IN the data)

VR Data Analysis

- Spatial immersion and perspective
  - Inherently more intuitive understanding of relationships between variables
  - Natural evolution: raw numbers $\longrightarrow$ Screen visuals (seeing data) $\longrightarrow$ World visuals (being IN the data)
- Real-time data interaction
  - No command-line interface
  - Previews available

VR Data Analysis

- Spatial immersion and perspective
  - Inherently more intuitive understanding of relationships between variables
  - Natural evolution: raw numbers $\longrightarrow$ Screen visuals (seeing data) $\longrightarrow$ World visuals (being IN the data)
- Real-time data interaction
  - No CLI
  - Previews available
- Higher-dimensional plots
  - Few suitable implementations outside of VR
- Dynamic data
  - Few suitable implementations outside of VR

Overview of ROOT VR

- Developed with Unity engine
- Runs as an executable
- Is launched automatically from within the ROOT session
- Links to headset via Meta Link

Implementation: High-Level Review

- ROOT::VR exists as its own **namespace**
- **Abstract** VRHist class (analogous to TH1)
  - Derived VRHist1, VRHist2 template classes

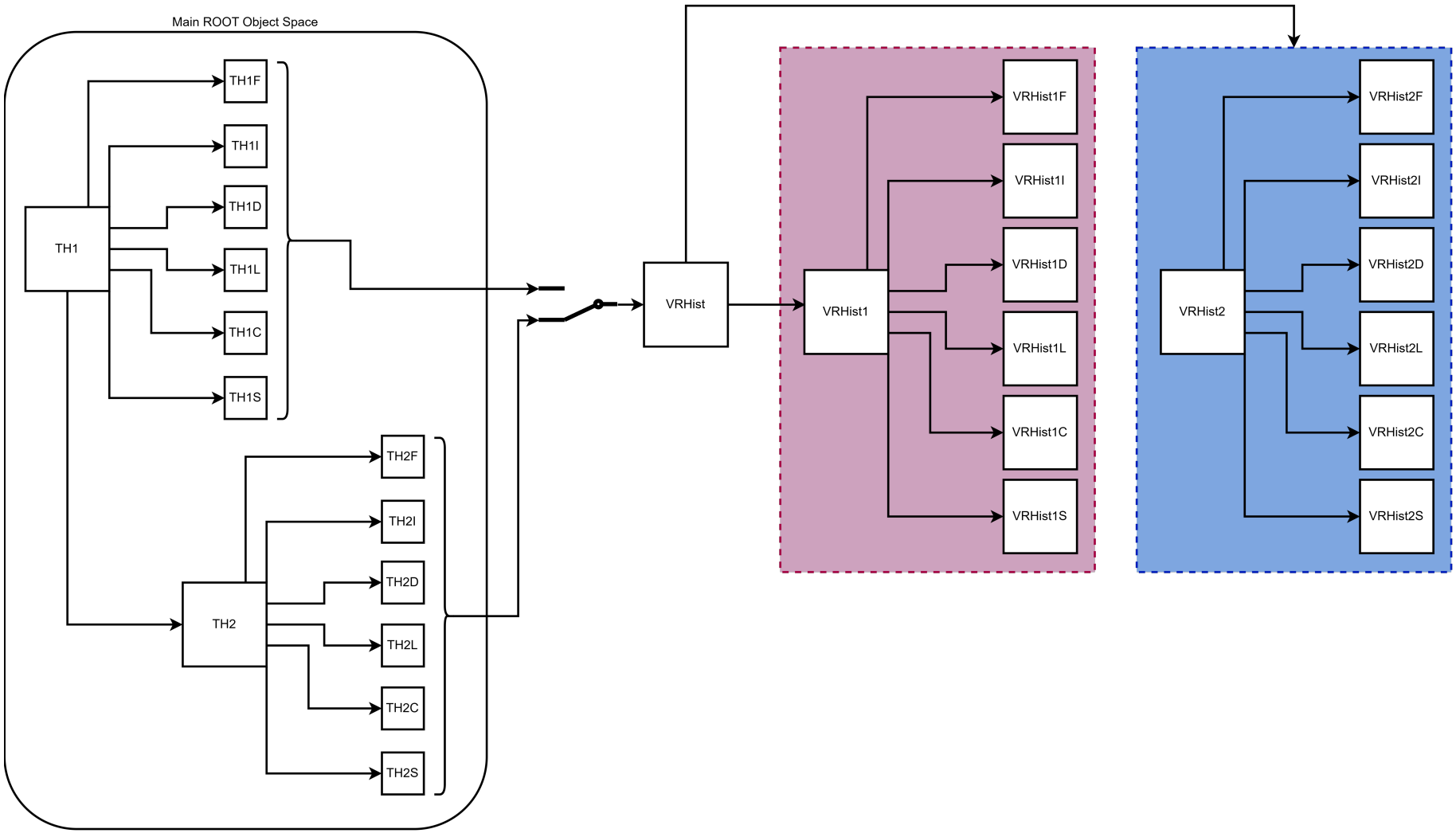

Figure 5: Inheritance structure of ROOT VR

UCLA

Implementation: Data Details

- ROOT communicates with Unity through **NamedPipe**
  From hist/hist/VR.cxx

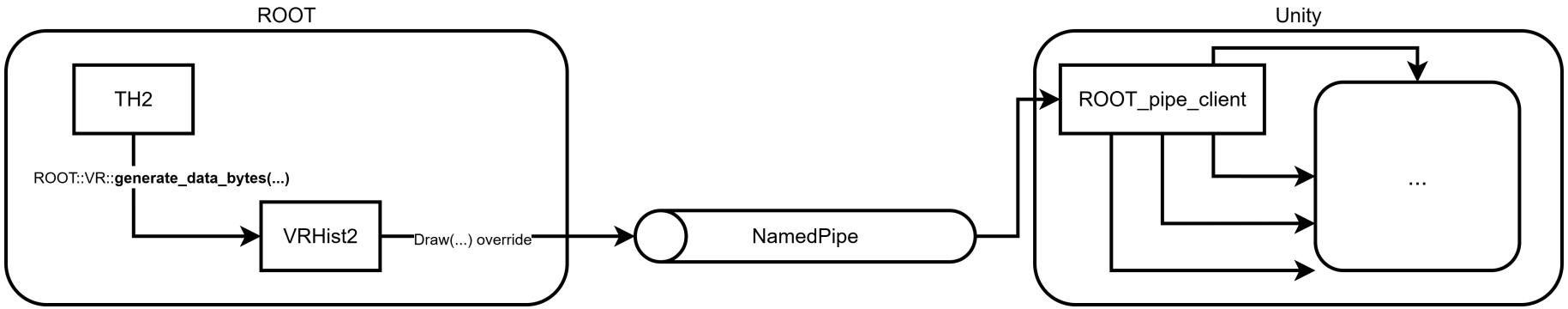

Figure 6: Data flow in ROOT VR

Implementation: Unity Implementation and Optimizations

- LEGO bars are scaled and transformed cube meshes
- Creation, indexing[1], placement, scaling of bars are all delegated to the GPU

```
Graphics.DrawMeshInstancedIndirect(...)
```

[1] "Indirect" in the function name distinguishes it from a similar function which does not index instances.

Future Plans: Higher-Dimensional Plots

- Data is modular; can be easily updated to include high-D plots

```
enum THn { TH1, TH2, /*Future: TH3, TH4, etc.*/};
```

Future Plans: Dynamic Data

- See time-dependent data evolve in real-time
  - Instead of looking at data as a function of time on a static graph, see real change in data over time
- Move the scaling from **Start()** function to **Update()**, which runs every frame

Future Plans: Overlays, ROOT Options

- Overlay TH2 colour plot (figure 7)
- Integrate more of the existing ROOT **Draw(...)** options, e.g. colours, surfaces, etc.

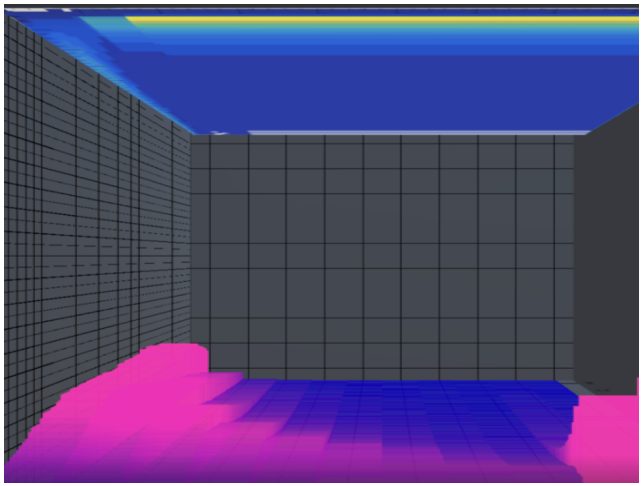

Figure 7: TH2 colour overlay on VRHist2

Conclusion

ROOT VR aims to achieve all goals of data analysis in virtual reality:

- Spatial immersion and perspective – ✓
- Real-time data interaction – ✓
- Higher-dimensional plots – Future
- Dynamic data – Future