

CMS Event Display through the Oculus Rift

Kai Wisser - South Dakota School of Mines and Technology

Advisor: Souvik Das - University of Florida



Objective

The Oculus Rift allows the user to be immersed in a virtual reality where they can move around and interact with their surroundings. By integrating the Oculus with ROOT and CMS analysis software (CMSSW), it can be used to view events of interest that happen within the CMS detector. It will allow scientists to view events up close at all angles including a view from within the center of the event. It may also be used for physics outreach by the CMS experiment.

Steps of Oculus Integration

The steps of Oculus integration involve:

- Connecting to the Oculus Head Mounted Display (HMD)
- Retrieving head tracking information from the HMD
- Stereo projection to the HMD

The detailed process is fairly complicated and involves stitching together OpenGL, ROOT, CMSSW, and the Oculus API.

Stereo Rendering

The stereo rendering of an image splits the image into a left and right view that are offset from the center by 32mm each to account for the 64mm average distance from pupil to pupil, with each image slightly rotated as it would be seen by each eye. To do this in ROOT we had an event display system that we patched to display the detector in stereo side by side as required by the Oculus Rift. Specifically in TGLViewer.h/.cxx and TEveViewer.h/.cxx. With the addition of the flag:

```
Bool_t quad_buf = kTRUE
```

These changes allowed ROOT to take in stereo rendering from OpenGL.

To render the Fireworks display in stereo view we had to point CMSSW to the thus patched version of ROOT. Once built with the new ROOT, the CMSSW source code is altered so that the wrappers around TGLViewer and TEveViewer allow for the stereo functions from the two to be called by FW3DView, the class responsible for the image in the 3D Tower view of CMSSW. The class structure of CMSSW can be seen in the Figure.

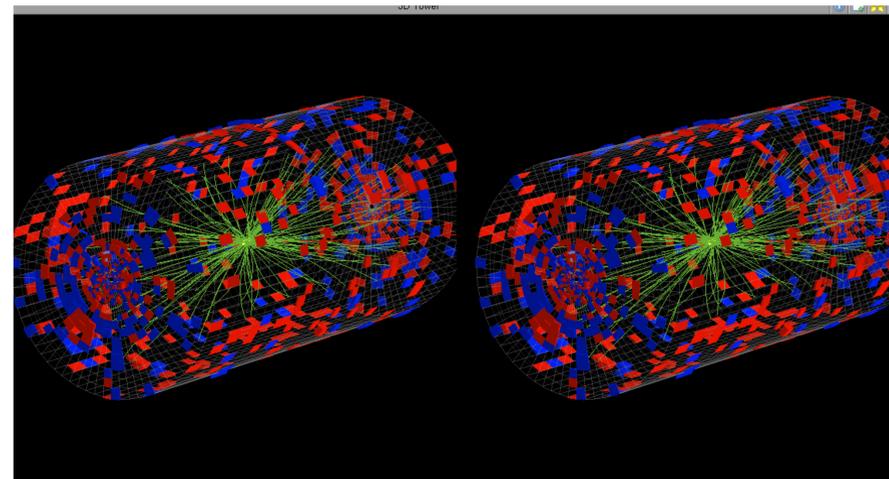


Image: Stereo rendering of CMSSW Fireworks display for the Oculus Rift

Head Tracking

For head tracking, we retrieved the HMD's orientation in 3D space as a quaternion which is of the form:

$$\mathbf{q} = w + xi + yj + zk$$

where i , j , and k are an imaginary set of axes with a fourth axis of 1, making \mathbf{q} a four dimensional vector representing rotation and orientation. We then derived the conversion of a quaternion orientation to the Euler Angles (roll, pitch, yaw).

These Euler Angles specifying the orientation of the user's head will be input as event interrupts in CMSSW in order to update the camera orientation within the viewer.

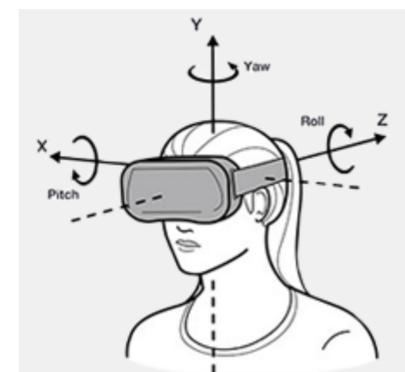


Image: Oculus Euler angle orientation

Future

While the Oculus HMD can now be used to view events projected with CMSSW, there are a few more steps required before its use becomes seamless. These are:

- Head Tracking Implementation
- Distortion Correction

The basic code used to connect the Oculus HMD to ROOT's Event Handler will be patched into the code controlling the camera orientation and mouse movements for the 3D Tower view.

Distortion correction is done frame by frame and is a recurring function that occurs in every frame rendering. It utilizes Brown's Distortion Model on each color channel to fix both pin-cushion distortion and chromatic aberration caused by the lenses of the Oculus.

Acknowledgments

Alja Mrak Tadel - Matevz Tadel - Patrick Gartung

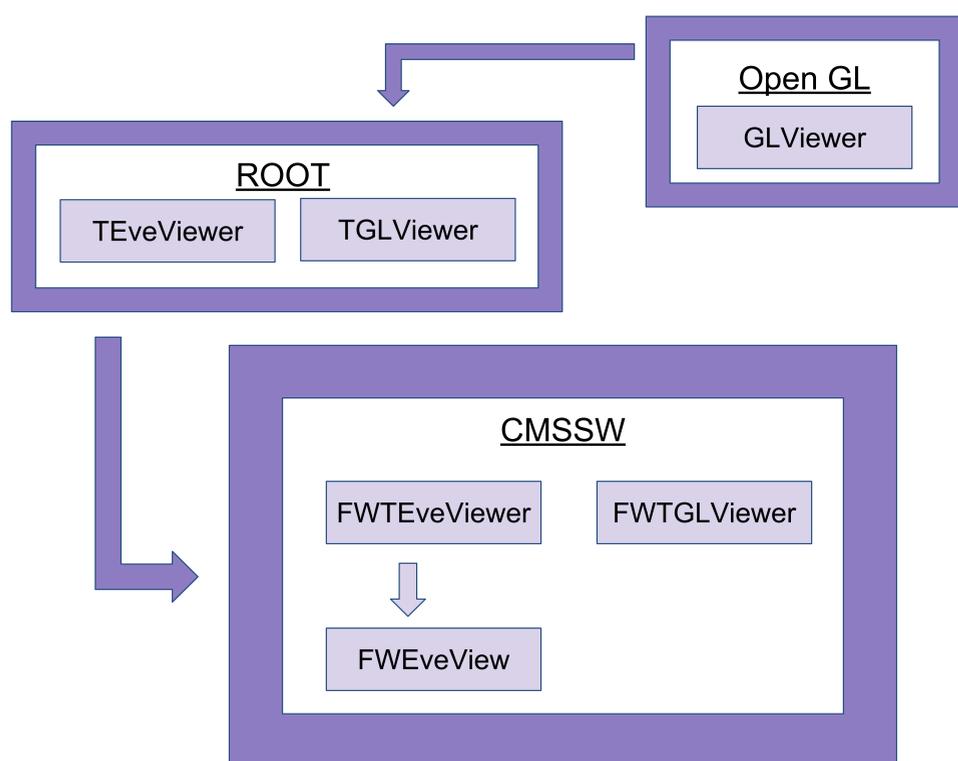


Figure: Structure of class inheritance from OpenGL to ROOT to CMSSW when dealing with the classes modified for stereo view.