

sam bilbow

projects writing engagements

polaris~

Project Presentation: TEI '21 Student Consortium

Project Outline: TEI '21 Student Consortium Paper

[← Hardware](#)[Software: Running Demos →](#)

Software: Calibration (December – May 2021)

Initiation

Originally, my system was very temperatmental. After much deliberation, I realised I must have shorted some part of my integrator. After replacing it, everything is plug and play. Luckily I was mainly working on written parts of my PhD during this period of time.

For best results I plug in the USB, and then the DP cable into my computer, and unplug them in the opposite order.

Software Development Kits Needed

Intel Realsense SDK (enables head tracking)

- Latest Intel RealSense SDK 2.0 – you're looking for an asset file like "Intel.RealSense.SDK-WIN10-2.47.0.3309.exe"

Ultraleap Gemini SDK (enables hand tracking)

- Leap Developer Kit 5.0.0-preview+52386

- [Leap Developer Kit 4.0.0+52238](#) - need to rollback to this driver to run Leap Motion example demonstrations such as [Paint and Cubes](#)

Optical Calibration

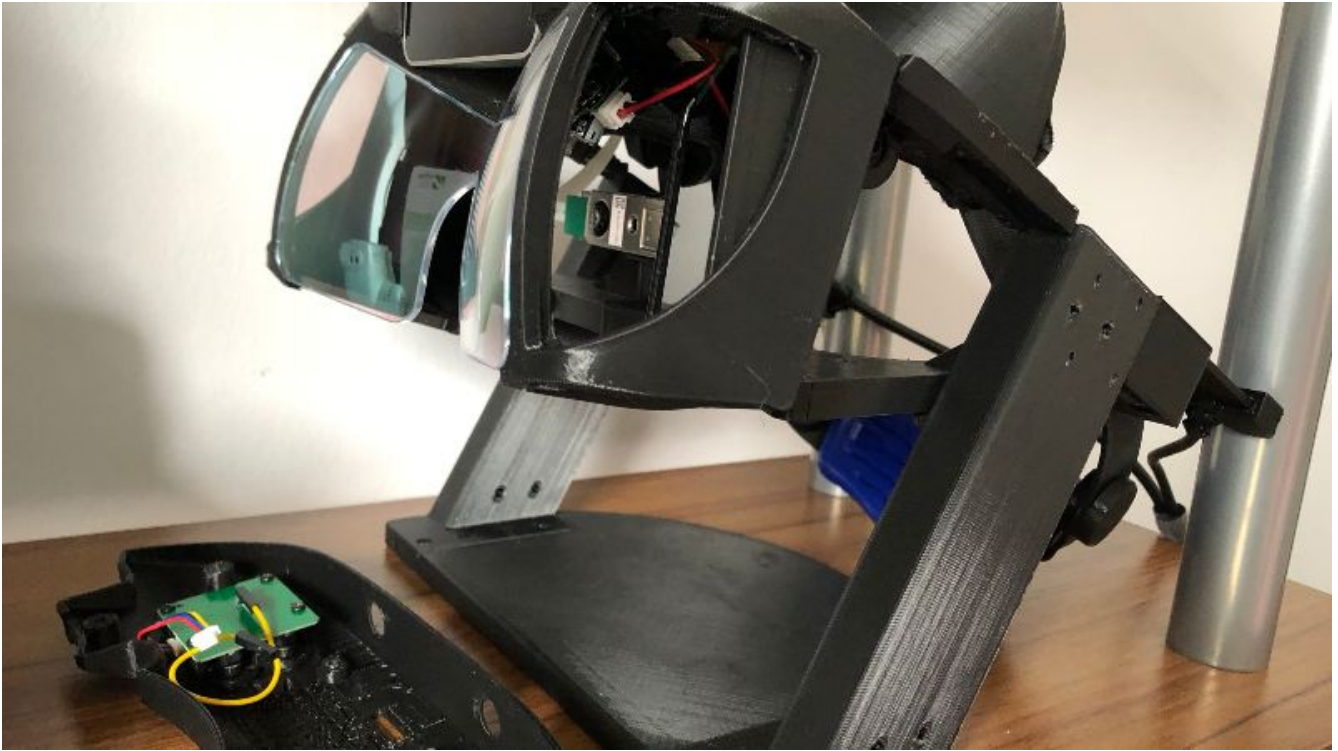
Calibrating the headset is necessary due to the large amount of small variables between headsets this stage. Lets have a look at some things that might not be the same for everyone:

- 3D Filament Type (material and layer height)
- 3D Printer Tolerances (how accurate the print is +/- mm)
- Resultant Inter-Hardware Position (the above two factors might result in minute differences in position between hardware components)

For this reason, it is commonplace to "calibrate" your headset through software, by creating a file which contains information about the various positions of hardware (I believe mainly the screens). This already sounds very difficult, so how does it work?

Historically, there are two different types of calibration, 3D and 2D. 3D came first, and is therefore (frustratingly) sometimes referred to as V1 calibration. 2D came second, and is referred to as V2 calibration. See the differences [here](#).

- 3D Calibration (V1) requires the use of two extra cameras and is therefore more expensive and a longer process. It was necessary to do this before the North Star had a 6DoF sensor with a dual-camera at its disposal
- 2D Calibration (V2) uses the dual-camera in the 6DoF sensor (Intel T261), and is therefore cheaper and quicker.



Historical 2D (V2) Calibration Stand vs New T26x Hybrid Calibrator

2D (V2) calibration used to require placing the headset on a special calibration stand, taking the T261 and anchoring it where the eyes would sit behind the headset. These are the first photos in the slideshow.

Thanks to the new **T26x Hybrid Calibrator**, the stand is no longer necessary, as is shown in the last two photos

Calibration is fairly simple and requires running premade python scripts. [Instructions are here](#) (I created a custom anaconda virtual environment (venv) to install the packages and run the calibration scripts from this venv). The output are four number arrays, which can later be used in Unity to make sure that each eye receives calibrated visuals from the screens. I keep these arrays in a sacred folder called "CALIBRATIONS DO NOT TOUCH"

[← Hardware](#)

[Software: Running Demos →](#)

Resources

Headset Documentation: Project North Star

Community: Project North Star Discord Server

Repository: Project Esky Renderer