**Pupil-Labs Analysis Steps**

*note – you may need to run dos2unix.py on camera pickle file if the following error occurs: ‘*UnpicklingError: the STRING opcode argument must be quoted’

Participant data is large - 20GB per participant.

So is held in the 'Archive Backup' hard-drive in the Driving Lab, and also on Callum's work external hard-drive. You will need to work from an external drive for the initial processing stage of the analysis. A copy of the raw data is also kept in the cloud.

**Step 1) Correct for distortion and save new surface definition file.**

Run extract\_markers\_CDM.py on a single recording folder with the ‘write’ option set to True. This will output an undistorted world.mp4.

What is in the surface\_definitions file? Checkl

Then load the folder into pupil-player and correct the surface definitions. Check that all the markers are being used. If they are not you might need to make them bigger in your experiment file.

Close pupil-player. A new surface definitions file will have been saved. Copy this somewhere safe.

Re-run extract\_markers\_CDM.py with the write option set to False (as we don’t have the space for lots of new world.mp4s). This will extract the undistorted marker positions you need and save it in a markers.npy file. This step will take a while.

Check that there is data in the markers & surface\_marker\_cache files.

**Step 2) Obtain gaze positions relative to the surface.**

Run main.py. This will output a csv file for each individual, in the rootfolder of that individual.

**Step 3) Collate gaze position csvs and move processed data to github analysis repo.**

Now we want to run analyse\_gazeonsrf.py. This translates the csvs of normalise positions onto angular estimates on the surface and saves a big csv of all files for analysis in R.

**UPDATED ANALYSIS PIPELINE: 29/07/19**

**STEP 1 – Recalibrate**

Using the pupil dump files, run **multicalibrate.py** to recalibrate offline and produce pldata files.

On the office PC the python scripts need to be ran from Anaconda prompt in the pupilenv. They cannot be ran from visual studio code.

**STEP 2 – Extract the undistorted marker positions.**

We want to build up a cache of the undistorted marker positions. Run **extract\_markers\_pldata.py**

**STEP 3 – Create gaze\_on\_surface\_csvs**

Now using the recalibrated pldata we create csvs of a the normalised position on the surface. This undistorts the gaze and maps it to the surface. Using **create\_gaze\_csv\_pldata.py**

**STEP 4 – stitch with steering data**

The final stage is building a large data frame that stitches the gaze data with the steering data for each trial, and projecting gaze through into the world. The script that does this is **stitch\_and\_project\_gazeandsteering.py**

**Steps from raw to usable data.**

**Retrieve gaze -> undistort -> project onto surface -> project onto screen.**

**For th along midline analysis.**

1. Save\_th\_frompath to get closest point on midline and/or future path.
2. Gaze through midline densities to return maximum mode(s).
3. th\_to\_focal\_point.py for quick calculation of th to gaze modes.
4. Plot\_data.py to get midline\_cumdist. Needs to change.
5. Save data frame
6. Linmix.py for cluster modelling to estimate TH.
7. R multi-level modelling for doing TH inferences