# Progress Summary

## 22+23+24/4/2020

My objective this week was improving the overall performance of the DSO on our own dataset. The steps I took to accomplish that are:

1) Writing a python code that uses the UEYE interface, in order to record a video for calibration and also for recording with auto exposure. I found a code made by IDS that displays live feed from the camera, and learned how to use the UEYE interface correctly. Afterwards, I wrote an Object-Oriented code that takes 3 arguments:

    - **FPS:** the fps we want our camera to record with.
    - **Parameter set file:** the path to the parameter set file we want to use.
    - **Calib:** if its equal to 0, it records a normal video. And if it is equal to 1, its records video meant for the response calibration.

    **Example:** python3 captureVideoUEye.py --fps=30 --parameterSetFile="parameter_sets/ultimate.ini" --calib=0

    The reason for the calib argument is that I noticed, that in their dataset the video starts with exposure time ~0, and then it slowly increases until it reaches the max exposure time. So, I did that too and it greatly improved the response calibration (pcalib.txt). The pcalib.txt now contains values from 0 to 255 and it is monotonically increasing without having to change anything manually. Furthermore, this also improves the vignette calibration, since it relies on the pcalib.txt file (keep in mind recording the video for the vignette calibration is done with --calib=0).

    However, improving the calibration is not this code's main objective. Previously I wrote a code that also gets the images folder and times.txt file from the camera feed but the times.txt had constant exposure. So, this code brings the times.txt file way closer to their implementation where it takes to account the auto exposure and the time stamps are by the actual capture time of the photo.

**2)** Improving the parameter set file (camera configuration) in order to improve the DSO performance. I have changed and tested many different configurations in order to get better results, and found out that having the master gain cranked up a bit, helped the performance. However, the DSO still didn't perform on our dataset as it did on their dataset.

I believe, the problem is the resolution that we are using (640x480) in order to overcome the fact that we are using a fisheye lens. Another factor is, the camera calibration was done a chessboard printed on an A4 paper which is not ideal. So, it is best to wait for the new lens and new large chessboard to arrive.

**3)** Testing the DSO on a video where I walk in around a chair in the living room of my home. At first it didn't work, I believe it is because the factors I mentioned above. However, after I walked and turned slowly, the DSO recognized the PATH!

I have also tested the LDSO, and it gave a much better path!