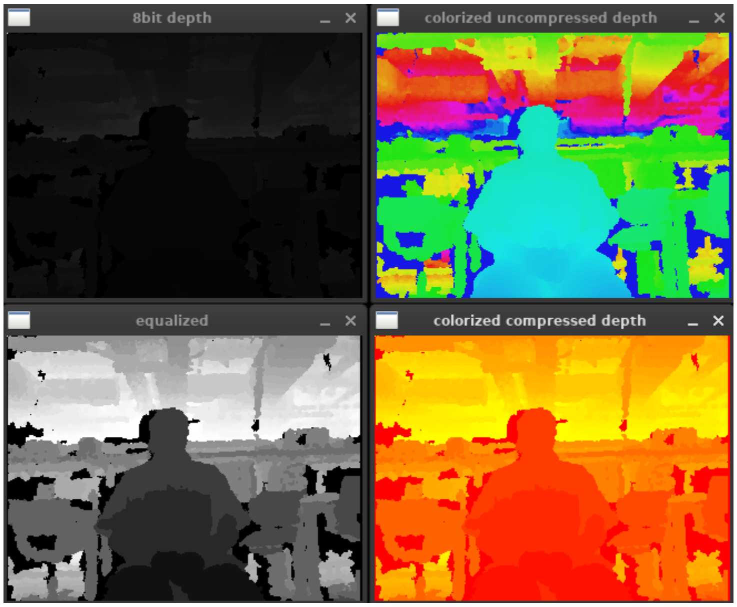
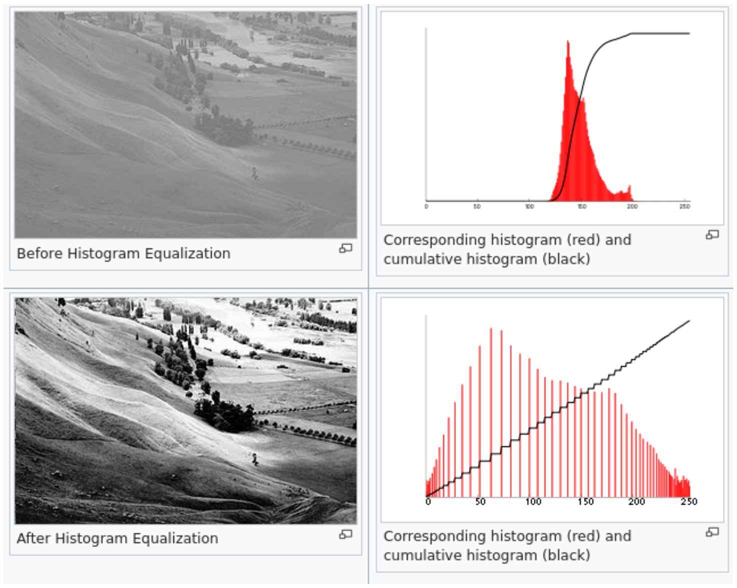
WEEK 4:
Hours: 12

The first thing I worked on this week was a set of four new contact microphones. To prepare these sensors for the use of the team, I desoldered the original wires that came with the disks and soldered on thicker, longer wires. This treatment will make it easier to route the wiring when we start attaching hardware to the full table.

The rest of the week was spent trying to come up with a way of visualizing the depth data that retained the entire range of depth values. This may or may not help certain tracking algorithms find the ball more easily. I tried several different methods:



In the top left panel of the above figure, one can see the result of simply truncating the last 8 bits of the 16-bit depth value. The contrast is comparable to the data coming directly out of the camera. In the bottom two panels are the same 8-bit image after applying a process known as “histogram equalization”.



Before Histogram Equalization

Corresponding histogram (red) and cumulative histogram (black)

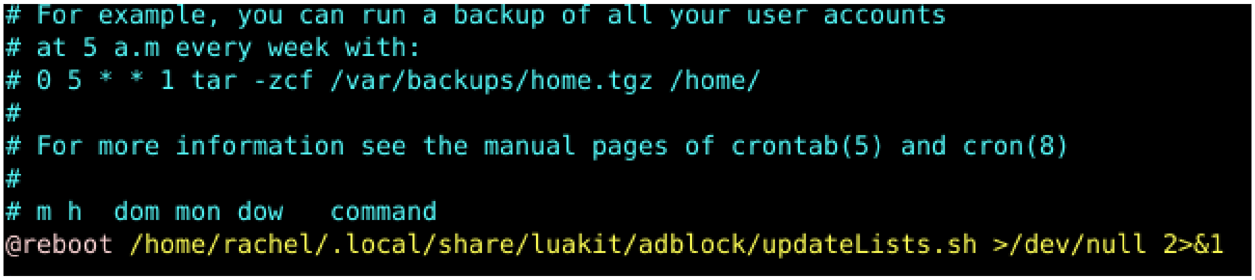
After Histogram Equalization

Corresponding histogram (red) and cumulative histogram (black)

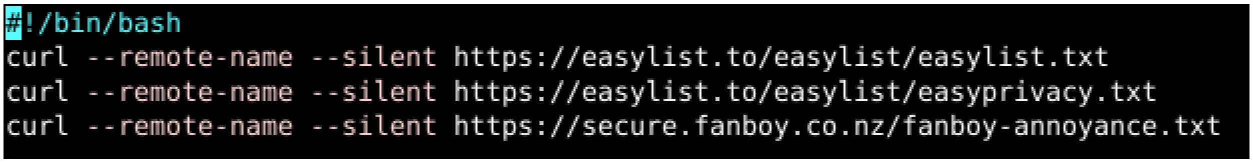
Histogram equalization works by stretching out the dynamic range of an image in such a way as to make the CDF of the pixel brightness histogram a roughly straight line, as can be seen in the bottom right panel of the figure above, taken from [1]. It's clear from the images on the left that this process can seriously improve the contrast of an image.

Finally, the top right panel is a version of the same image which retains the full precision of the 16-bit depth value by using color instead of brightness to denote the depth. This effect was achieved by mapping all the possible values of a `uint16_t` to a hue angle in the range 0-360, setting saturation and value to 0.9, and then converting the resulting HSV color into RGB. In the figure above, the colors wrap around 8 times in the 16-bit range, which would indicate that the depth sensor is capable of measuring depths much farther away than one can see in the lab where the photo was taken.

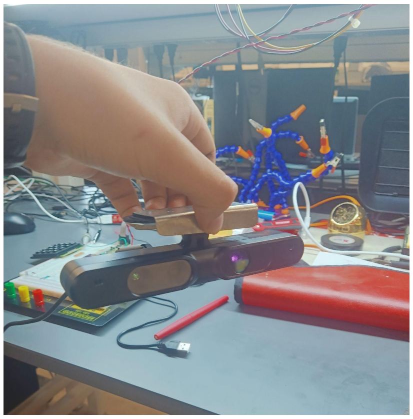
Also this week, I installed ad blockers on the main laptop or the project, which may seem irrelevant to the project, but I believe this benefits the project because ads make websites load much more slowly. Many of the websites from which I am learning how to use OpenCV are littered with ads which monopolize the very limited memory of the ThinkPad T420 which is being used for all the graphics. Saving time that I would have otherwise spent waiting for ads to load is clearly beneficial to the team. The blocklists are stored in text files which are automatically updated every time the computer reboots by a cron job which I set up for the task.



```
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
@reboot /home/rachel/.local/share/luakit/adblock/updateLists.sh >/dev/null 2>&1
```



```
#!/bin/bash
curl --remote-name --silent https://easylist.to/easylist/easylist.txt
curl --remote-name --silent https://easylist.to/easylist/easyprivacy.txt
curl --remote-name --silent https://secure.fanboy.co.nz/fanboy-annoyance.txt
```



In order to mount the depth camera over the table in a way that makes it easy to remove and adjust, I made a bracket out of sheet metal which holds the camera with a friction fit, shown below. The bracket is bent sheet-metal construction, with mounting holes.

[1] https://en.wikipedia.org/wiki/Histogram_equalization