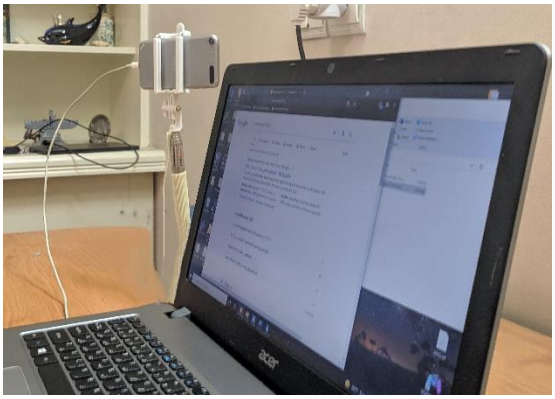
My DIY Camera Setup

I decided to create my own stereo camera setup at home in order to speed up my work, since it was difficult to have constant access to the ping-pong table.
I used my phone as one of the cameras, and my laptop's webcam as the second. Both capable of filming at 30 FPS (similar to the two webcams we currently have at hand)



I also printed a chessboard on an A3 paper, in order to be able to calibrate the two cameras for far distances

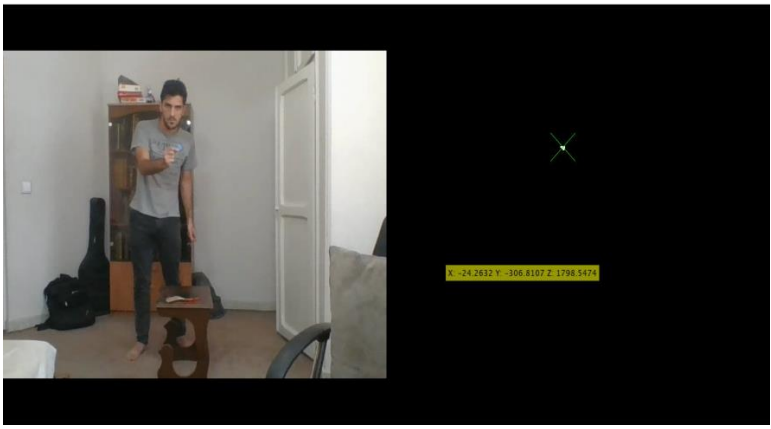
Note that since the ping-pong ball I had was white, which was difficult to track, I colored it to be blue which is much easier to track.

Calibration

I used around 50 chessboard image pairs taken at around 3-4 meters distance (in order to simulate a similar conditions of a ping-pong table). By doing this, I would be able to find the coordinates of objects up to this distance.
The calibration resulted in a mean Error of around 0.58, which is pretty good considering the number of image pairs.

The Results

after performing calibration for a farther distance, I managed to confirm that my code is able to show the real world coordinates for near distances, as well as far distances.



Testing How distance Impacts the world points:

I created a simple test to see if there is any drastic difference between the world points calculated for when the ball is distant to the camera, and when It is near the camera:
I considered two scenarios, one is that I'm near the camera and holding the ball with my arm

holding the ball back, and then I opened my arm to measure how much the depth changed:

| Z2 - Z1 | = 715 – 319 = 396

The second scenario, is the exact same as the first one, except that the ball is in a 4 meter distance from the cameras:

| Z2 - Z1 | = 2165 – 1770 = 395

As can be seen, it seems that the distance we plan to be working on, doesn't have much of a negative impact on the depth.

Measuring Speed of the Ball:

I added the speed measurement to my code. The way this works is that for each current frame, we find the Euclidian Distance between the current frame's world points and the previous frame's world points (this measures in Pixels Per Frame). We can also find velocity across each of the three main axis by finding the distance of their respective axis.