Mika Thormann 5465950

Progress report for week 15

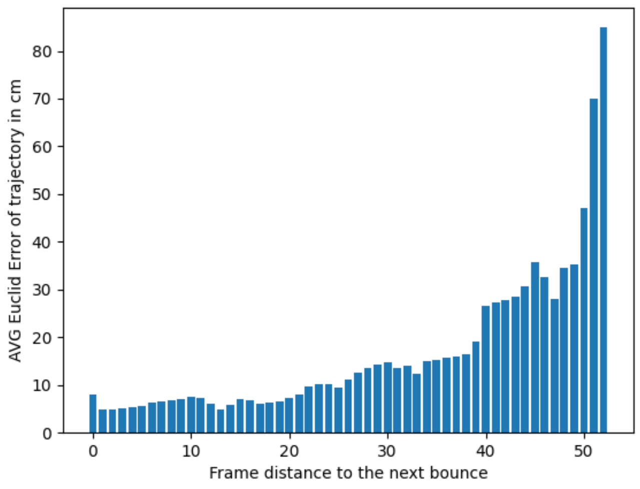
Differential Evolution and Benchmark:

I did a lot of testing and benchmarking. For that, I changed my simulation data generator to produce random trajectories in a speed range of 0 to 20 m/s, but only store reasonable ones which fulfill the following criteria:

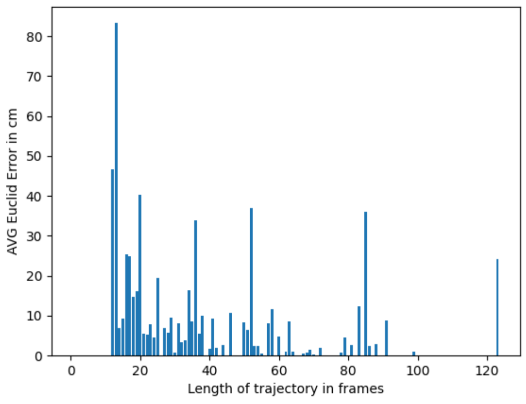
-min. 1 bounce and max. 2 bounces

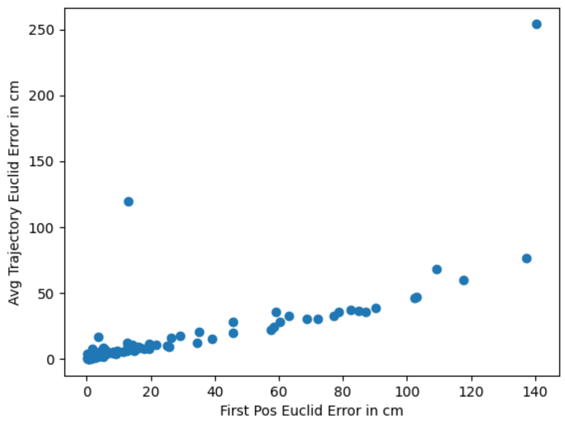
-hit the enemy side of the table at least once

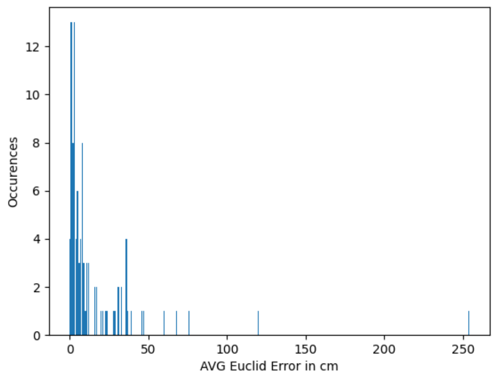
With these criteria, I still cover a wide variety of start positions, velocities and therefore angles. While testing, I added many more graphs for things that might be interesting, allowing me to find problems with the algorithm. In general, it works quite well on a low error most of the time, but there are some heavy outliers. The projection error is almost always really small, so most of the time a large error is caused by the predicted trajectory being moved in the z axis of the camera viewing direction. It seems impossible to deal with this problem, unless I restrict possible trajectories to be ones that are not against the rules of table tennis(hit the enemy side of the table). This is, because a bounce gives the trajectory a fixed point in 3D space which it will have to cross, limiting the amount of room for error drastically. This statement is supported by the fact, that even with this restriction, the accuracy drops for frames as they get further away from bounce-frames.

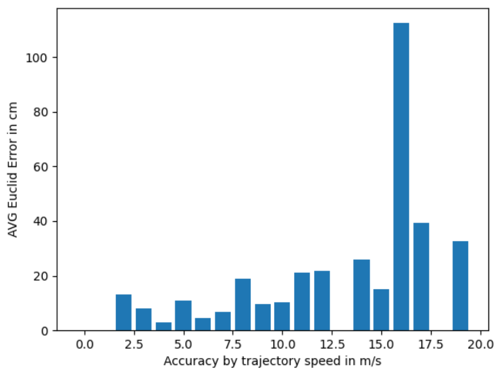


Still, there are some heavy outliers, which I will try to spot the reason for and fix.

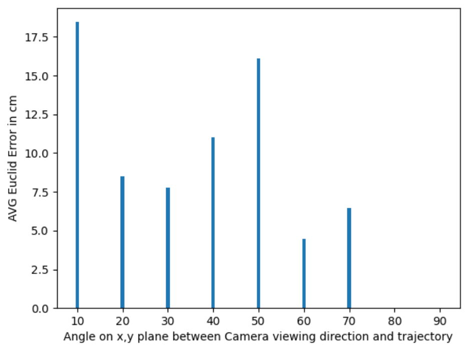


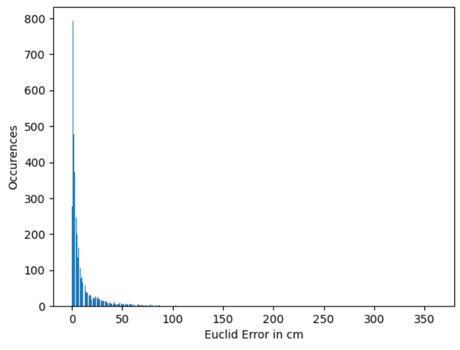


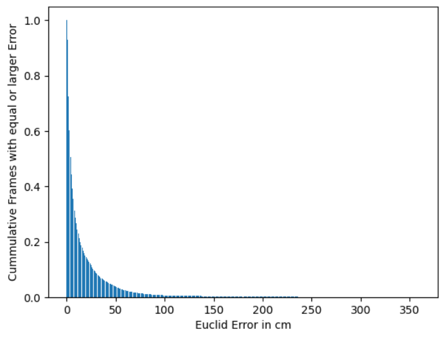


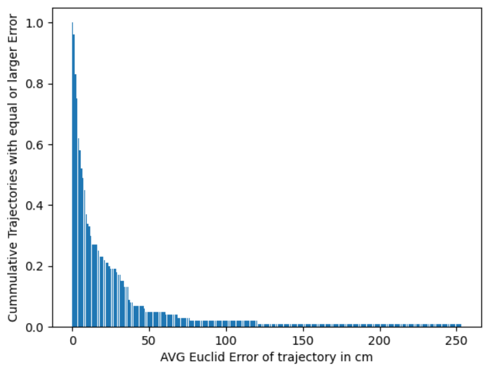


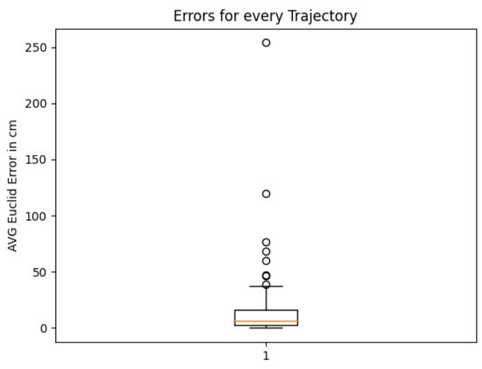
Below are more figures that you can use to get an idea of the current performance. The testing was done with 100 trajectories, generated by the initially stated method.

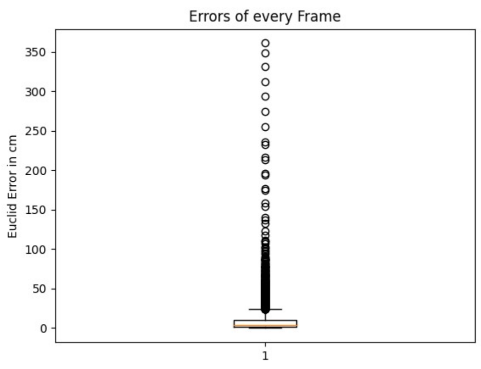












Other changes:

- I made the pose estimation more reliable
- I made the computation faster, by stopping when no improvement was made over a given amount of iterations

Swarm Optimization:

I tried swarm optimization and found that it converges faster, but yields worse results. That is why I stopped working on it and focused on improvingthe evolution algorithm.