Mika Thormann 5465950

Progress report for week 5

Tasks:

1) Implement TTnet (failed)

2) Change the file type for data to hdf5

2) Extracting the 3D ball coordinates from pose and 2D position

Implementation:

1) Failed

2) I now store the data in hdf5 files and created a new folder ball_data where the balls position and bounce-frame-indices are stored.

3)As the implementation of TTnet failed, I use generated trajectories from my physics simulation as data for the 3D extraction for now.

Given the approximated extrinsics and intrinsics, I can simply convert these coordinates to 2D with cv.projectPoints() and use them on calculated poses. Like that, I can even compare the original 3D points and the calculated 3D points to check if my code is working correctly. The following image illustrates how I can use simulated points and draw them into a given image.

To detect the bounce-frames, I simply check for sign changes.

For every frame that contains a bounce, I create a vector that leaves the camera through the 2D coordinate of the ball. Then I search for the intersection of that vector with the table plane(the table plane is already projected into camera space by using the extrinsics matrix) to receive the 3D bounce point in camera space.

What still needs to be done:

The next step is to use these points and the table's normal to generate new planes on which the ball is assumed to travel. With these, I can calculate the intersection for every frame in between(or every couple of frames, as the intersection calculation is costly).

The final step is to convert the accquired camera coordinates back to world coordinates.