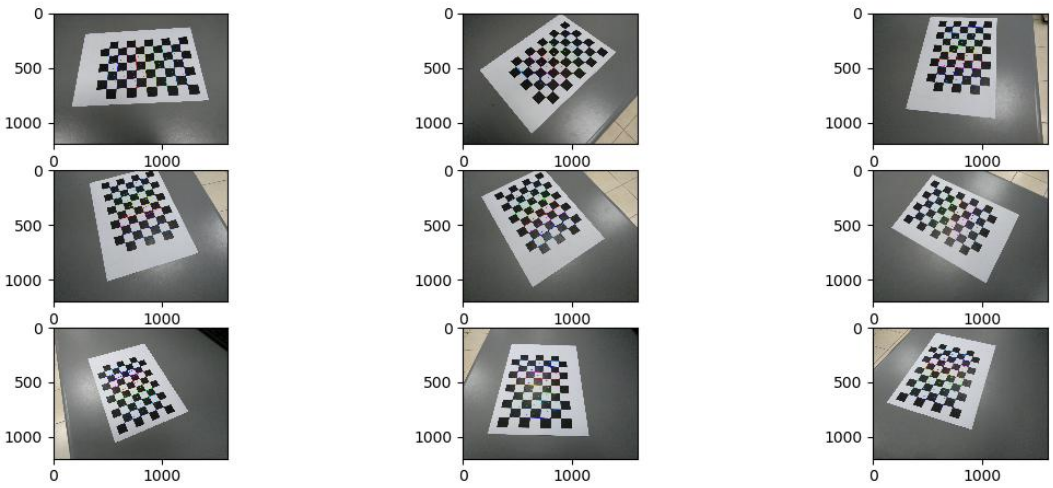
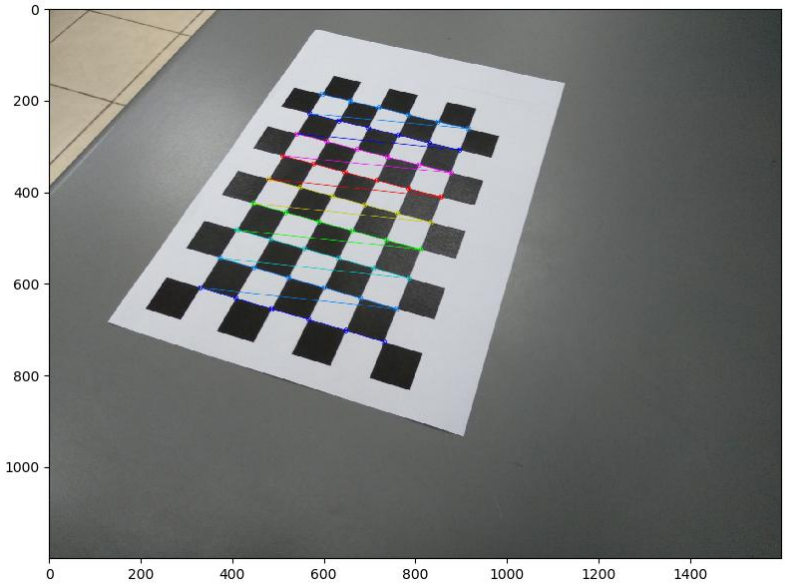
Project 2 – Augmented Reality

This project has multiple steps:

First step is performing the camera calibration in, here I've used a chessboard in different angels, as a reference, I've used the cv2.findChessboardCorners in order to find all corners of the given chessboard images, later I used the cv2.drawChessboardCornersin order to plot the following results:



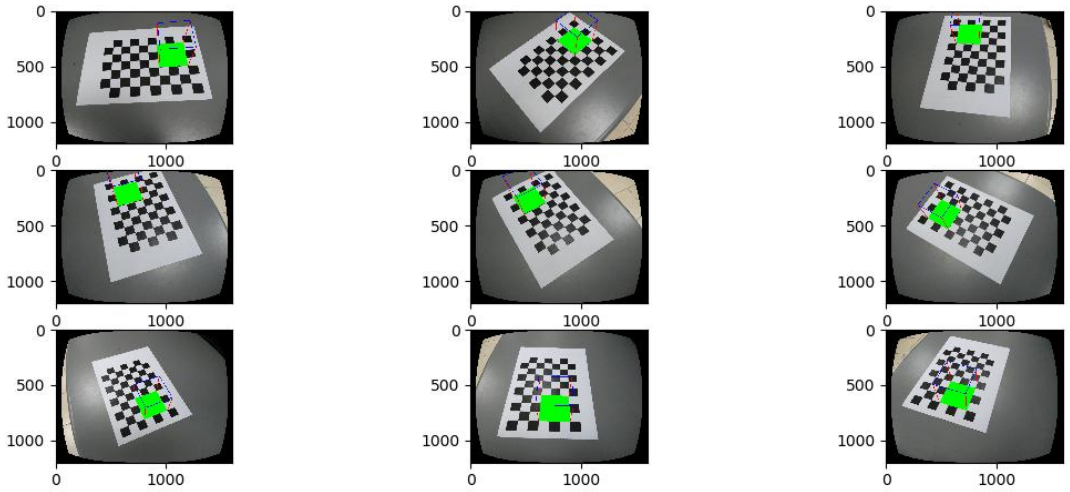
A closer look of the cv2.drawChessboardCorners:



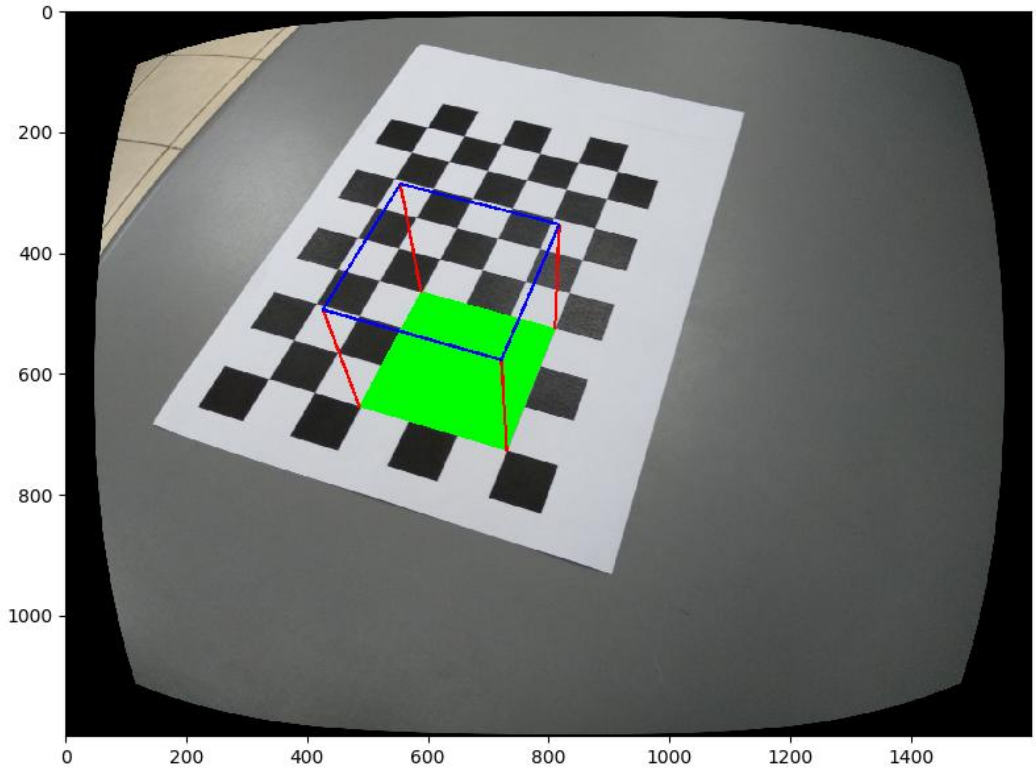
Later I've used the cv2.calibrateCamera in order to get the camera matrix and coefficients in order to create an .nbz calibration file that we'll use later.

Then I ran a loop on all images and ploted a rectangular shape on all of them, the result showed the high accuracy of the algorithm as showed the rectangure shape is located in the exact location in each image, this will come handy later.

Results:



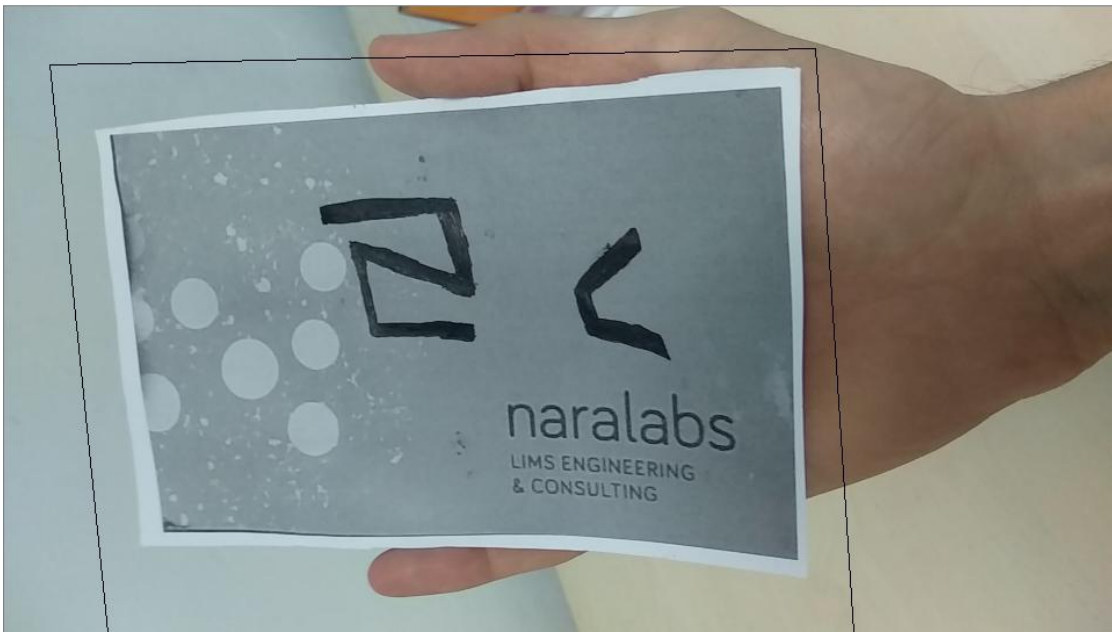
a closer look of the result(as you can see the images had been undistorted too):



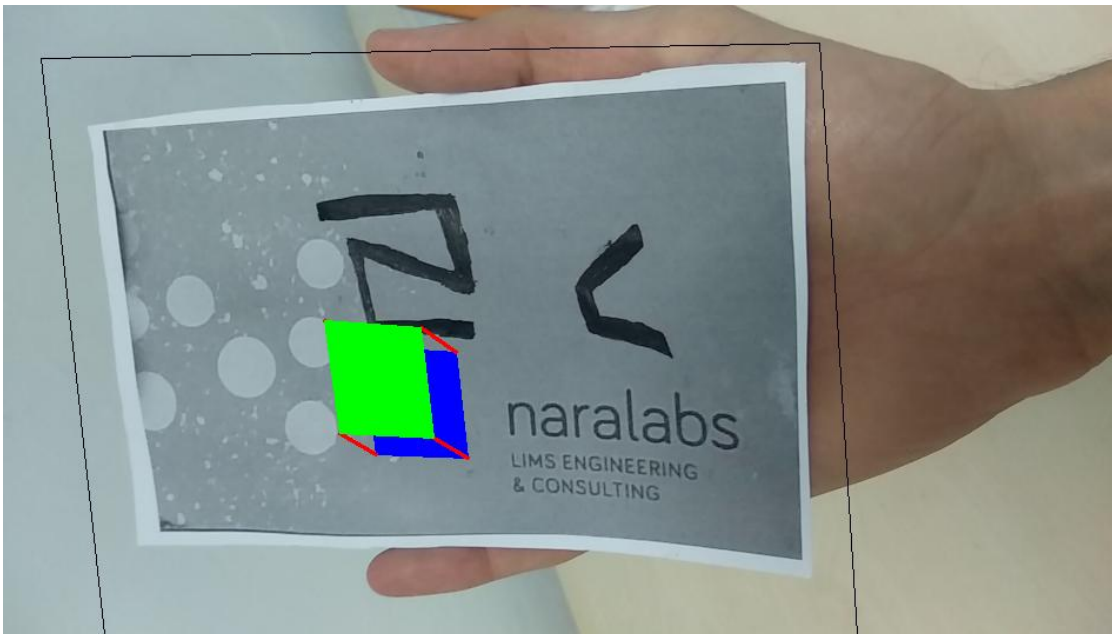
The next step is using BRISK/ SIFT feature detector, for some reason the SIFT didn't work, but the brisk yield a fairly good result, later I used the FLANN feature matching and Optical flow tracking and then camera pose estimation using cv2.solvepnpRansac and the final step is to project the object, in this case a 3d cube using the cv2.line function.

Results:

Cv2.polylines shows the region of interest the reference image:



Ploted 3d rectangular on the reference image:



Note:

During this project I was having trouble projecting .obj files, as a result I only managed to render a 3d cube, I tried to find a method to convert the obj files to a readable format that would allow the function to read the file, sadly that didn't work.