CS5330 PRCV - PROJECT 4 REPORT

Aadhar Bansal & Pranav K Nair

Description

(Calibration and Augmented Reality)

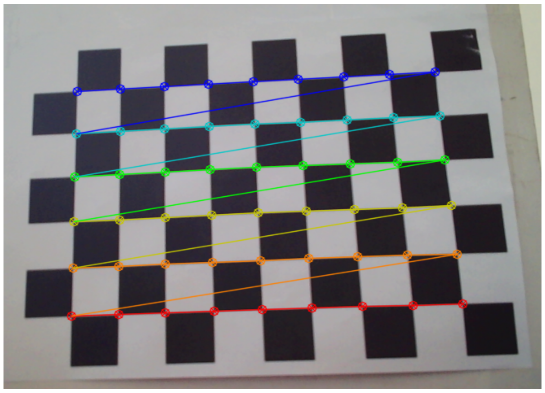
The aim of this project is to calibrate a camera and then use the calibration to generate virtual objects in a scene. The final result that we should achieve from this project is that the program can detect a target and then place a virtual object in the scene relative to the target that moves and orients itself correctly given the motion of the camera or target.

Tasks

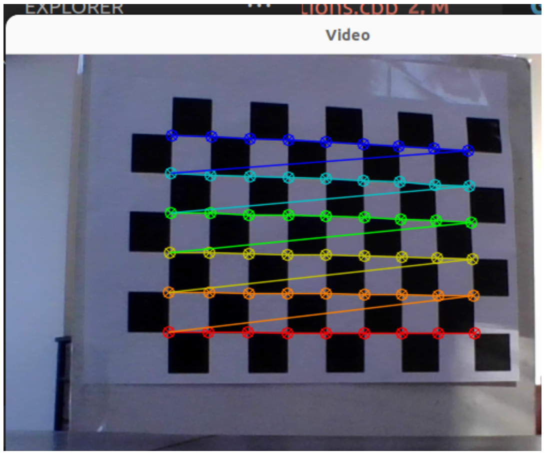
Task 1 & 2: Detect and Extract Chessboard Corners & Select Calibration Images

Corners of the Chessboard have been detected and the calibration images are selected

Camera 1



Camera 2

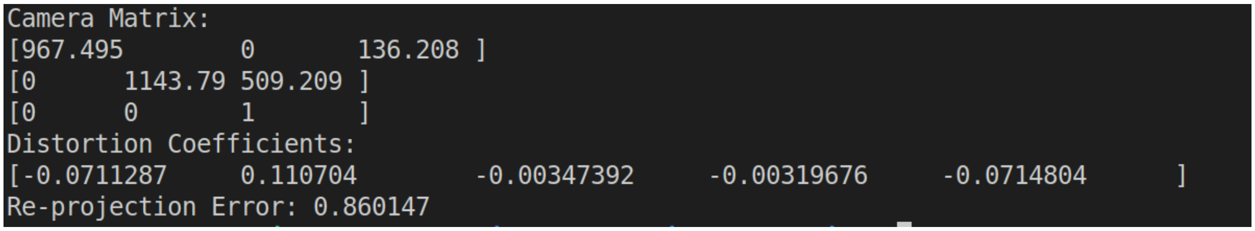


Task 3: Calibrate the Camera

Print out the camera matrix and distortion coefficients before and after the calibration, along with the final re-projection error. The two focal lengths should be the same value, and the u0, v0 values should be close to the initial estimates of the center of the image. Your error should be less than a half-pixel if everything is working well. For large images or cell phone images, the per-pixel error may be more like 2-3 pixels. Include the error estimate in your report.

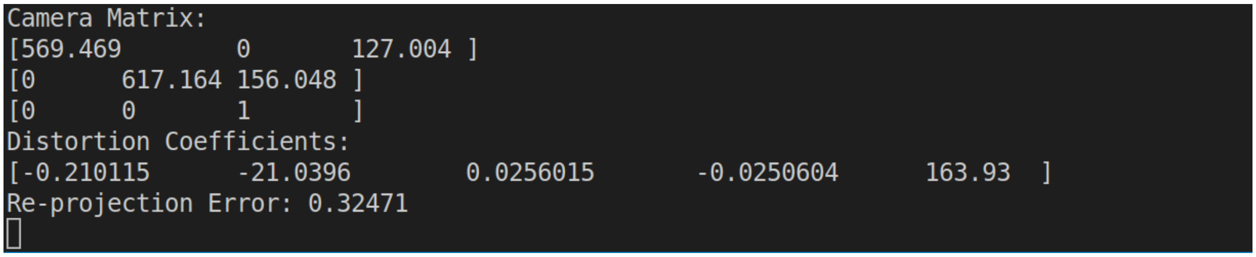
The camera calibration matrix, distortion coefficient and the error
calibrateCamera function provides the Camera Matrix and the Distortion coefficient

Camera 1:



```
Camera Matrix:
[967.495        0       136.208 ]
[0      1143.79 509.209 ]
[0      0       1       ]
Distortion Coefficients:
[-0.0711287     0.110704        -0.00347392     -0.00319676     -0.0714804      ]
Re-projection Error: 0.860147
```

Camera 2:

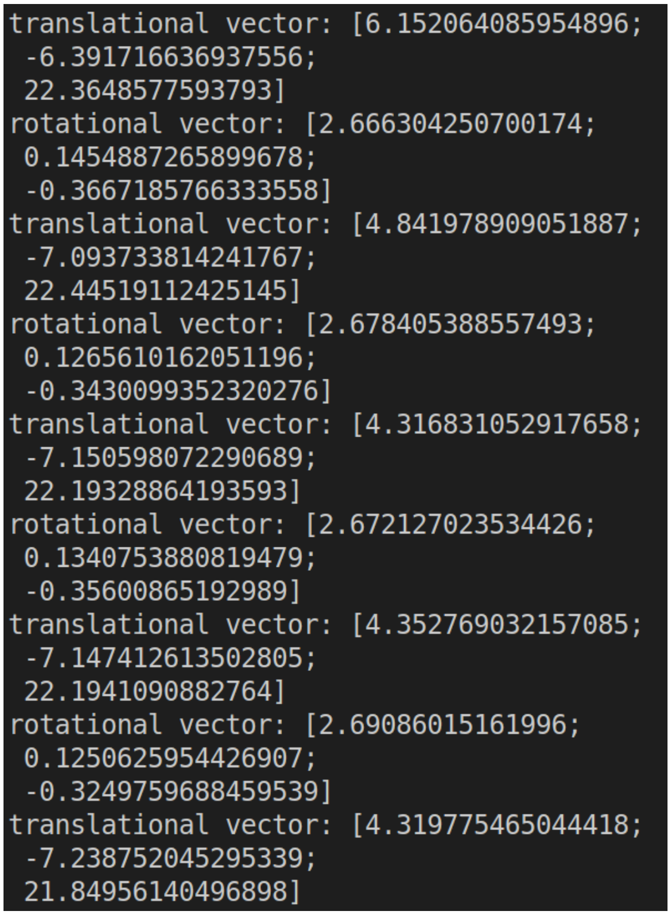


```
Camera Matrix:
[569.469        0       127.004 ]
[0      617.164 156.048 ]
[0      0       1       ]
Distortion Coefficients:
[-0.210115      -21.0396        0.0256015       -0.0250604      163.93  ]
Re-projection Error: 0.32471
```

As “Camera 1” has better resolution and was able to capture a larger image it throws more errors compared to “Camera 2” which is expected.

Task 4: Calculate Current Position of the Camera

The program prints out the rotation and translation data in real-time, as we were testing this task.



```
translational vector: [6.152064085954896;
 -6.391716636937556;
 22.3648577593793]
rotational vector: [2.666304250700174;
 0.1454887265899678;
 -0.3667185766333558]
translational vector: [4.841978909051887;
 -7.093733814241767;
 22.44519112425145]
rotational vector: [2.678405388557493;
 0.1265610162051196;
 -0.3430099352320276]
translational vector: [4.316831052917658;
 -7.150598072290689;
 22.19328864193593]
rotational vector: [2.672127023534426;
 0.1340753880819479;
 -0.35600865192989]
translational vector: [4.352769032157085;
 -7.147412613502805;
 22.1941090882764]
rotational vector: [2.69086015161996;
 0.1250625954426907;
 -0.3249759688459539]
translational vector: [4.319775465044418;
 -7.238752045295339;
 21.84956140496898]
```

Task 5: Project Outside Corners or 3D Axes

The corners have been highlighted by the circle (BLUE)

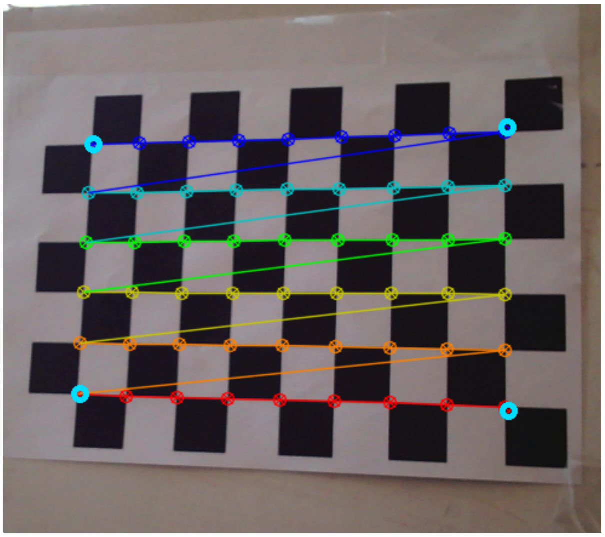
In this task these steps has been followed:

Projects 3D points onto 2D image plane.

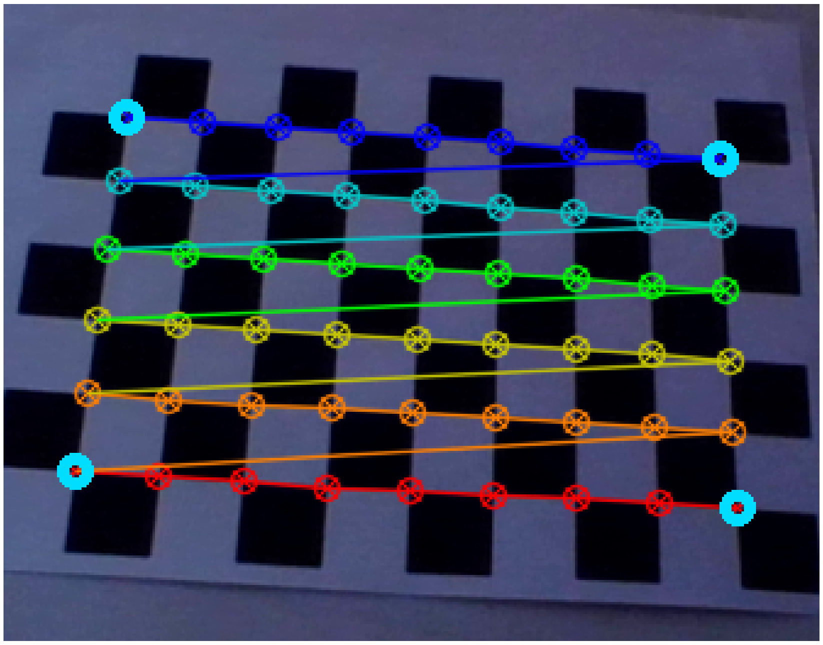
Draws circles on 4 specified corner points.

Uses camera parameters and distortion coefficients

Camera 1



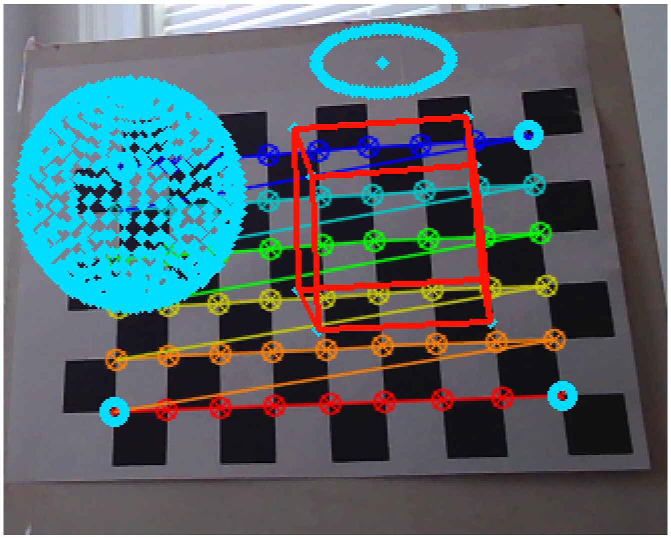
Camera 2



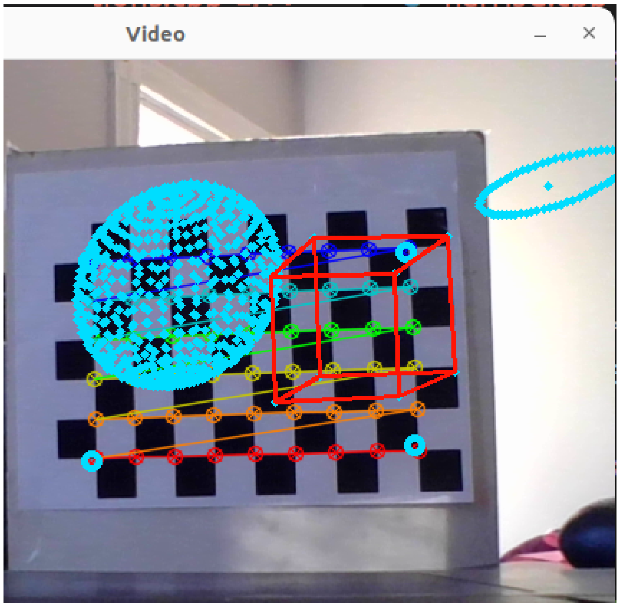
Task 6: Create a Virtual Object

We have created multiple creative virtual objects in the form of a cuboid, circle, and sphere.

Camera 1

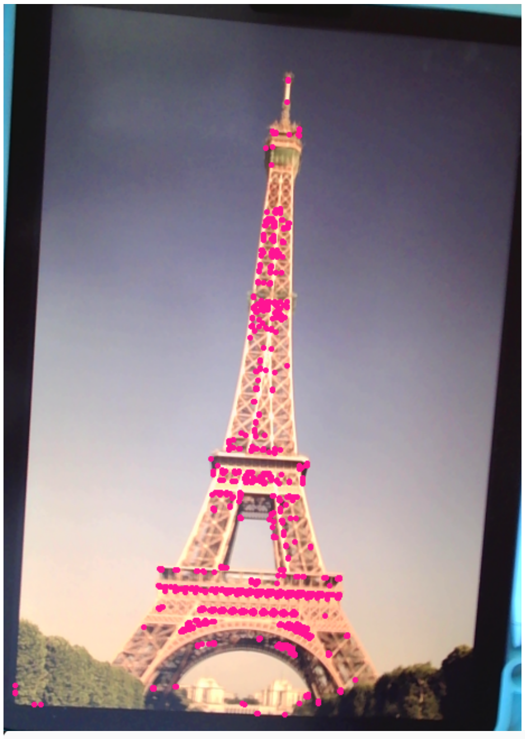


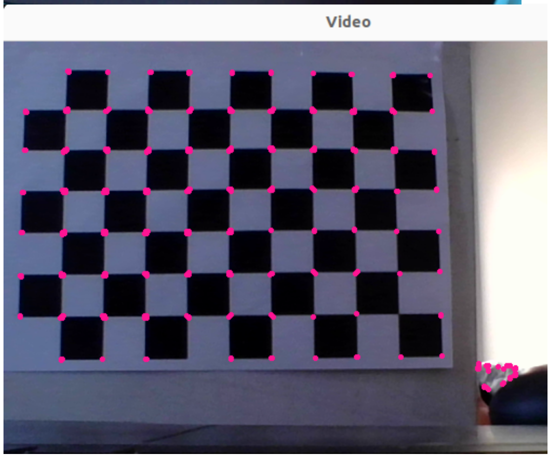
Camera 2:



Task 7: Detect Robust Features

We have implemented Harris Corner detection for this task.





Extension

Task 1: Test out several different cameras and compare the calibrations and quality of the results.

In the above tasks of the project, we have shown multiple outputs for each task. The outputs have been labeled "Camera 1" and "Camera 2" to show that we have tested the code with multiple cameras.

Task 2: Especially creative virtual objects and scenes are a good extension. It is possible to integrate OpenCV with OpenGL to render shaded objects (big extension).

We have added multiple creative virtual objects as seen in the output of Task 6.

Short Reflection

Aadhar Bansal:

In this project, i learned the various functionality and methods regarding the Camera calibration and the augmentation reality, this tasks included to get the camera calibrated using the chessboard and i have utilized the different camera so the calibration difference is also observed, then i explored the various functions of the openCV which helped in the augmented reality and learned how to project different three dimensional objects on the target.

Pranav K Nair:

I learned about the process of camera calibration and the concepts behind it. This was also the first time I worked on anything related to augmented reality, which was quite interesting.

Acknowledgment

Camera Calibration:
Opencv: https://docs.opencv.org/4.x/d4/d94/tutorial_camera_calibration.html

Camera Pose:
OpenCV: https://docs.opencv.org/4.x/d4/d94/tutorial_camera_calibration.html

Corner Harris:
OpenCV: https://docs.opencv.org/3.4/dc/d0d/tutorial_py_features_harris.html
Medium: https://medium.com/data-breach/introduction-to-harris-corner-detector-32a88850b3f6

Professor's Notes and the Textbook