NAME : JOSE JOSEPH THANDAPRAL

NUID NO. : 002102407

COURSE CODE : CS 5330

COURSE NAME : PATTERN RECOGNITION AND COMPUTER VISION

CRN : 36458

FACULTY : PROF. BRUCE MAXWELL

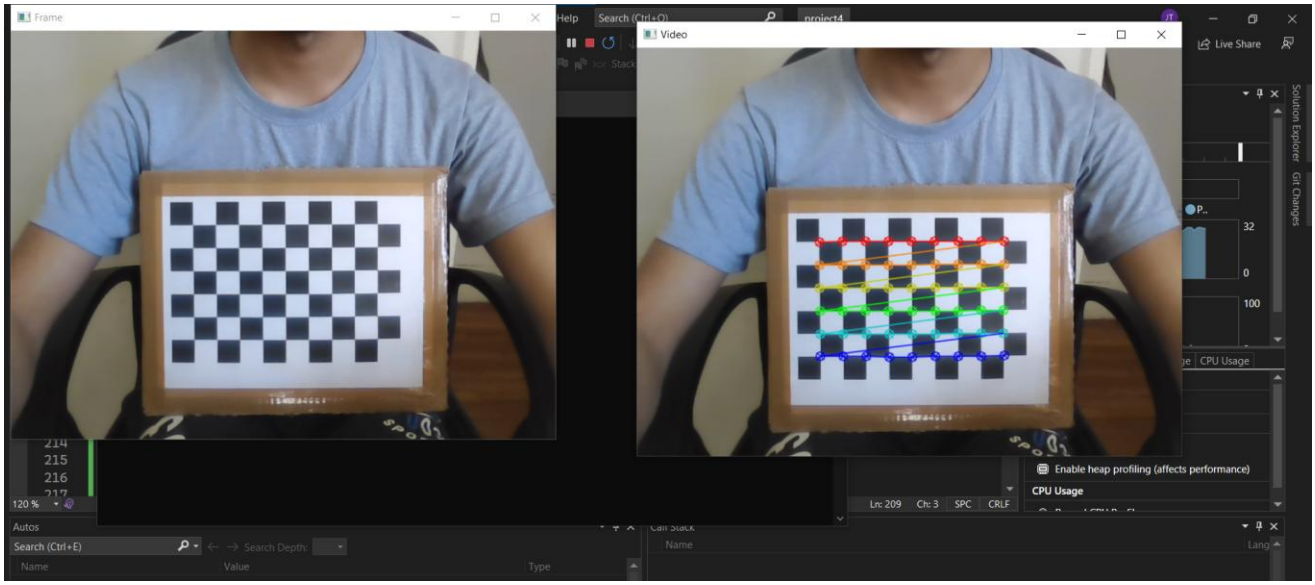
NORTHEASTERN UNIVERSITY, BOSTON

PROJECT 4

SUMMARY

The report includes the results and observations of the fourth project involving using the OpenCV package to calibrate a given camera and use the calibration parameters to augment virtual objects onto a checkerboard pattern. The checkerboard pattern is printed onto a surface and the user is prompted to hit ‘s’ as many number of time that the user wishes to save frames with the checkerboard pattern in it to go into the camera calibration function cv::calibrateCamera() as real-world object points and pixel-plane object points pair with correspondence between them, here the object being the checkerboard. Then the user hits ‘q’ which brings up the cv::imshow() window which show the checkerboard being tracked and drawn with green dots at its inner 5x8 checkercorner along with an axis at the first corner, as the user moves the checkerboard in front of the camera. Another cv::imshow() window shows an assymetrical object being augmented onto the checkerboard. Finally we use Harris corners as a feature being detected being detected on images which could be used as the basis for putting augmented reality into the image.

RESULTS: Task 1 & 2- Detecting and extracting chessboard corners and selecting calibration images.



Required Image 1: **a calibration image with chessboard corners highlighted**

Task 3 - **Error estimate:**

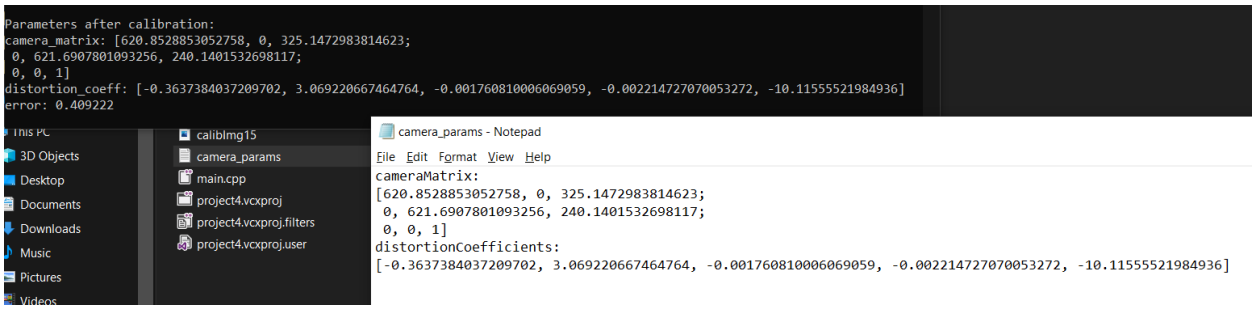
Before calibration:

```
Saved calibImg1.jpg

Saved calibImg2.jpg

Saved calibImg3.jpg

Saved calibImg4.jpg

Saved calibImg5.jpg

Saved calibImg6.jpg

Saved calibImg7.jpg

Saved calibImg8.jpg

Parameters before calibration:
camera_matrix: [1442.772093396958, 0, 329.4835716670173;
 0, 1448.099285217988, 255.3344198063892;
 0, 0, 1]
distortion_coeff: [-2.059703566388472, 158.1005503641477, -0.01733235085344272, -0.005546231301162283, -5675.197172671839]
error: 0.23389
```

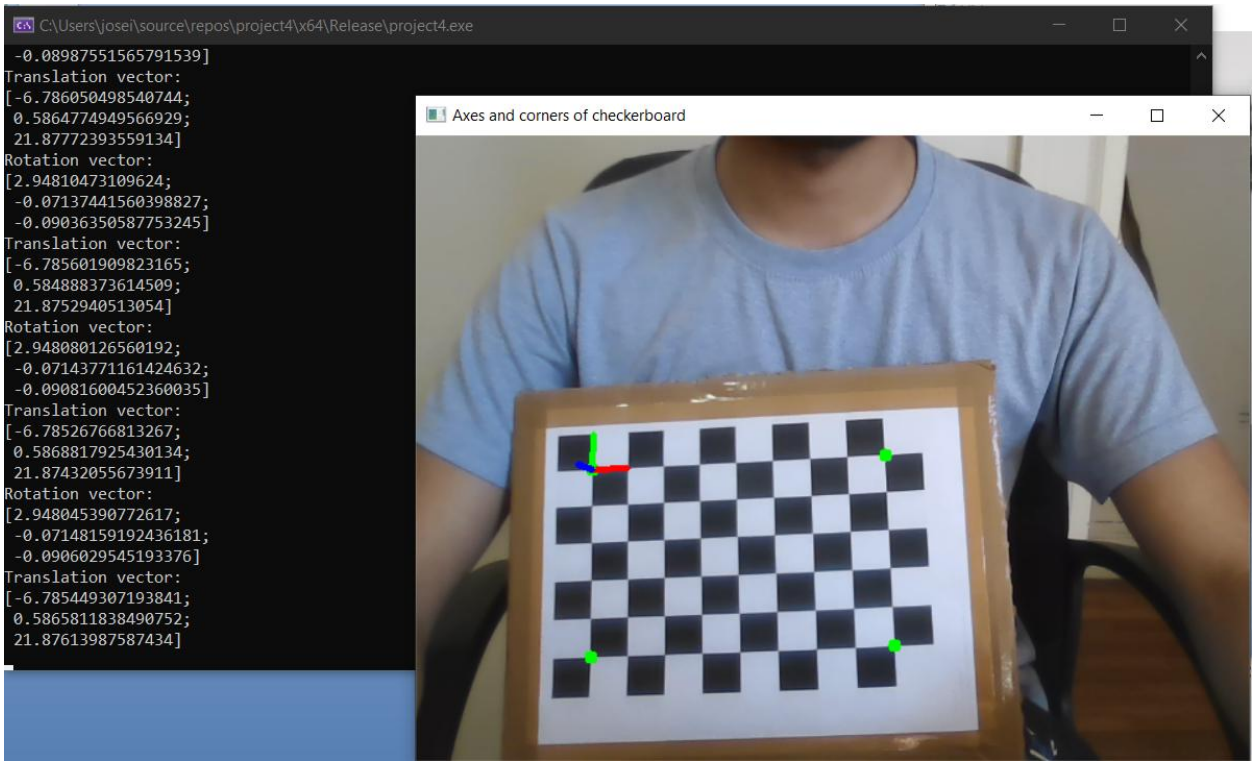
After calibration with **final error estimate**:

```
Parameters after calibration:
camera_matrix: [620.8528853052758, 0, 325.1472983814623;
 0, 621.6907801093256, 240.1401532698117;
 0, 0, 1]
distortion_coeff: [-0.3637384037209702, 3.069220667464764, -0.001760810006069059, -0.002214727070053272, -10.11555521984936]
error: 0.409222
```

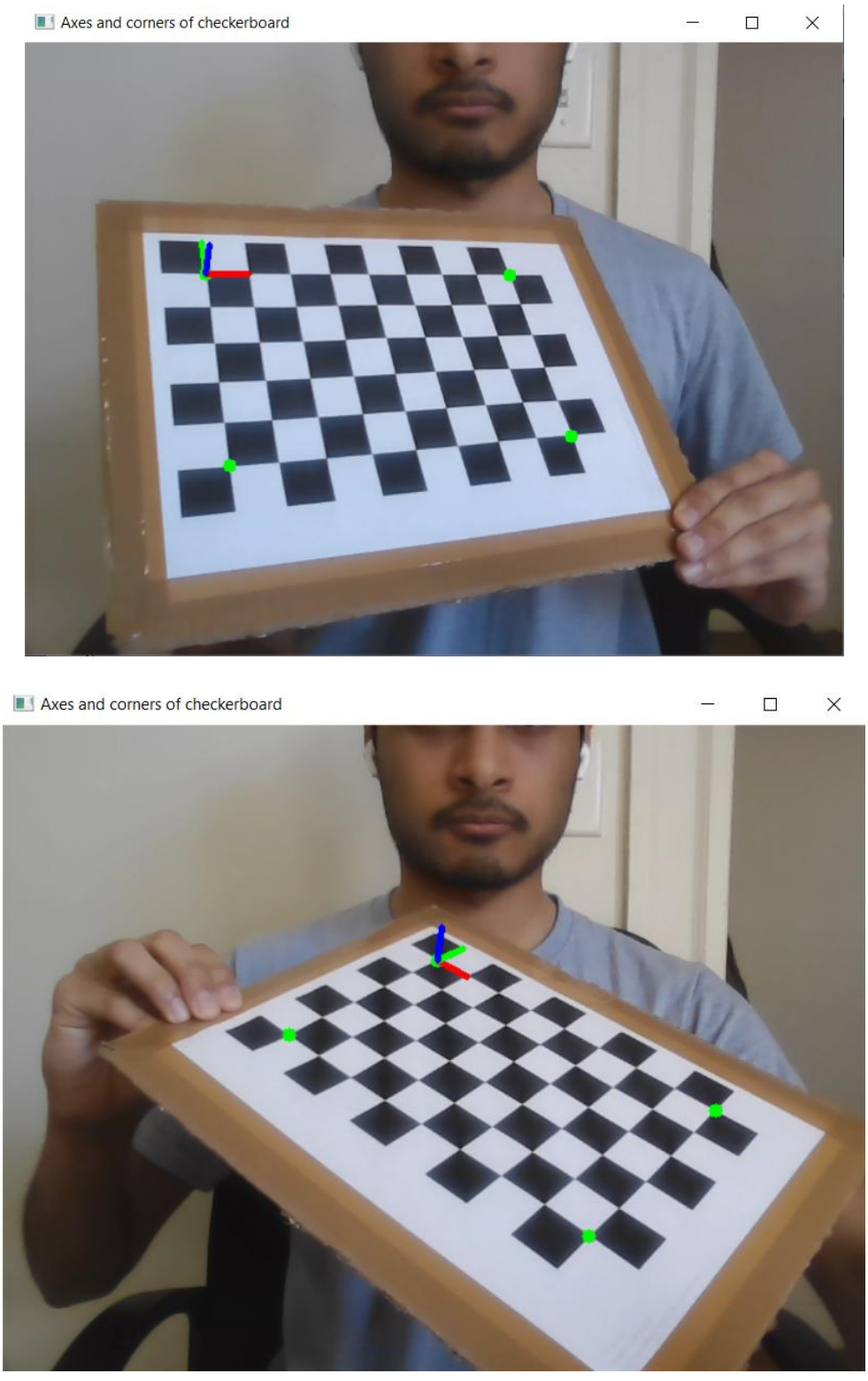
Writing camera matrix and distortion coefficients to file:



Task 4 - **Rotation and translation data printed on console in real-time:**

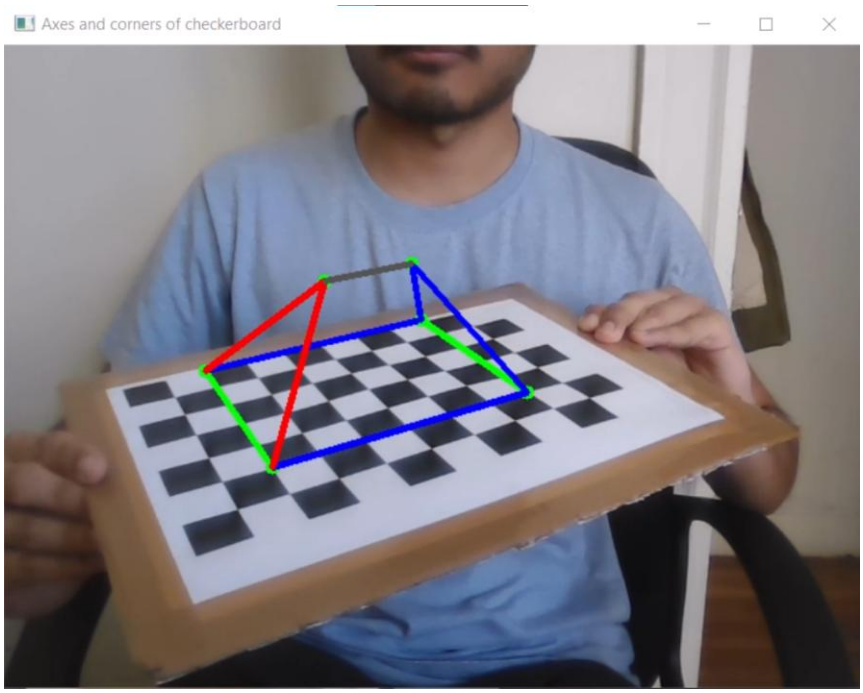


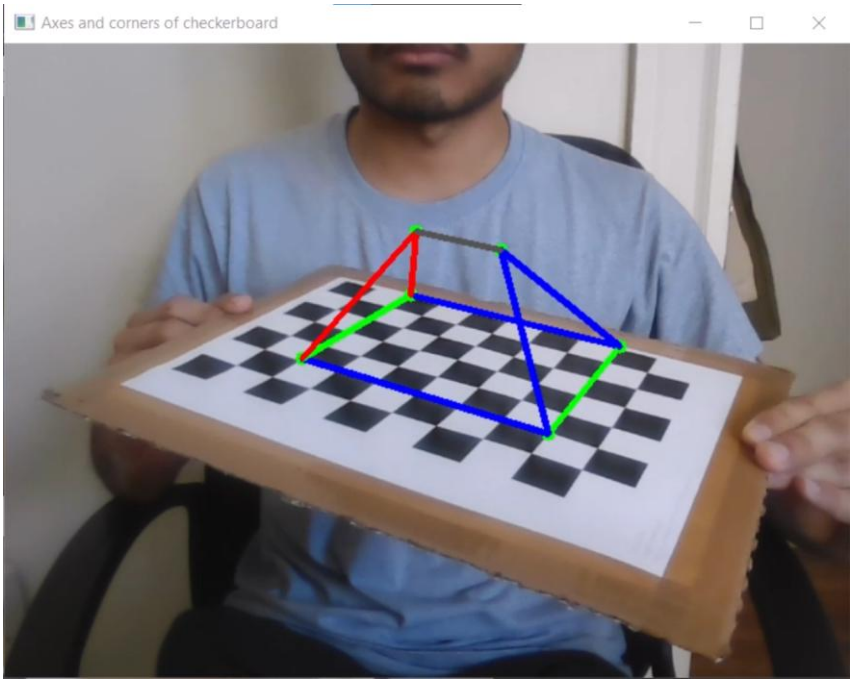
Task 5 – Project outside corners and 3D Axes



Required Image 2: Project outside corners and 3D Axes

Task 6: Create a virtual object



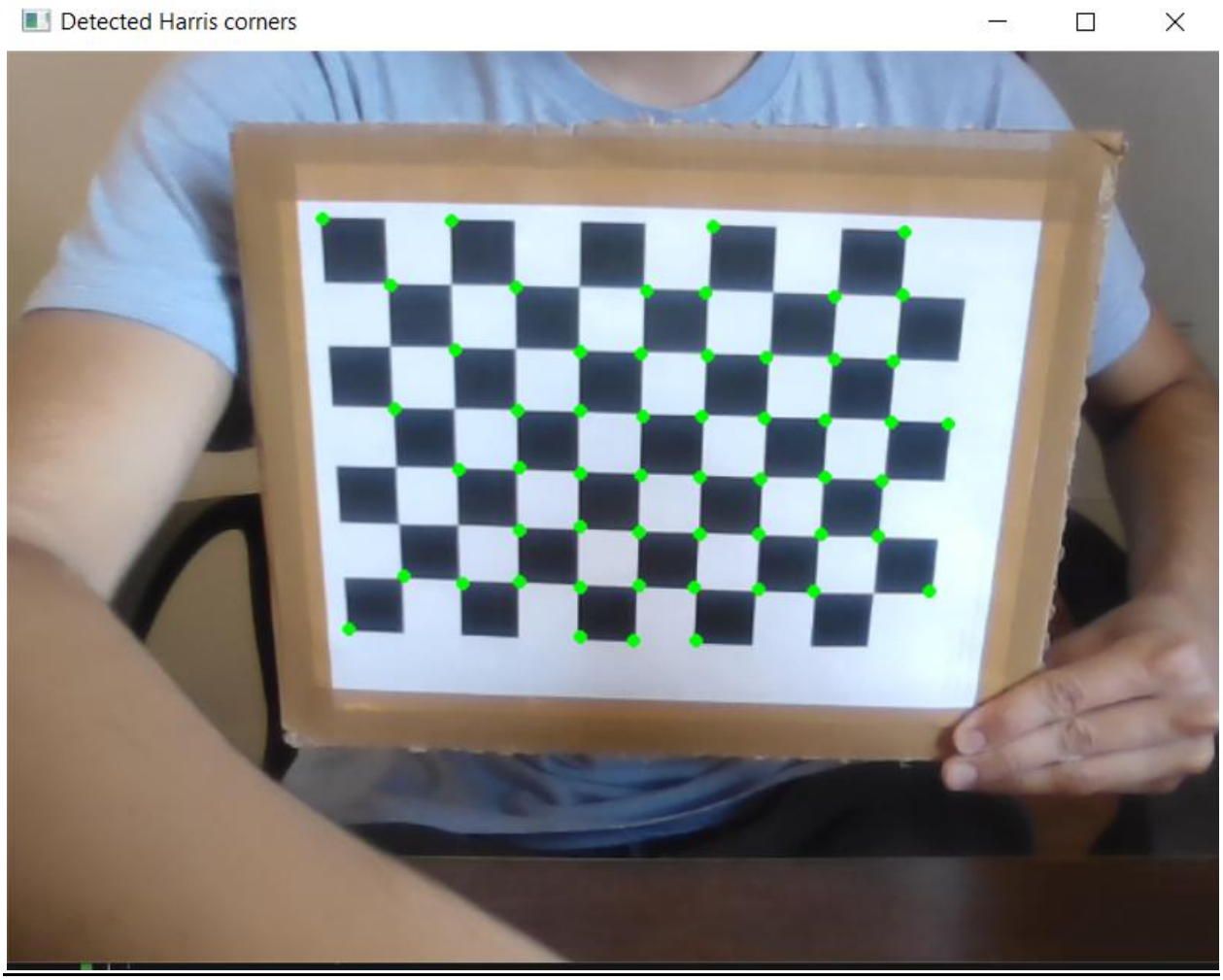


Asymmetrical triangular prism object with colored edges

Video links to 3D virtual objects:

https://drive.google.com/file/d/1qfEgv3UNKyIIsGfakPUDJqqx5QCjjMQj/view?usp=sharing – Asymmetrical triangular prism object with colored edges

Task 7: Detect Robust Features

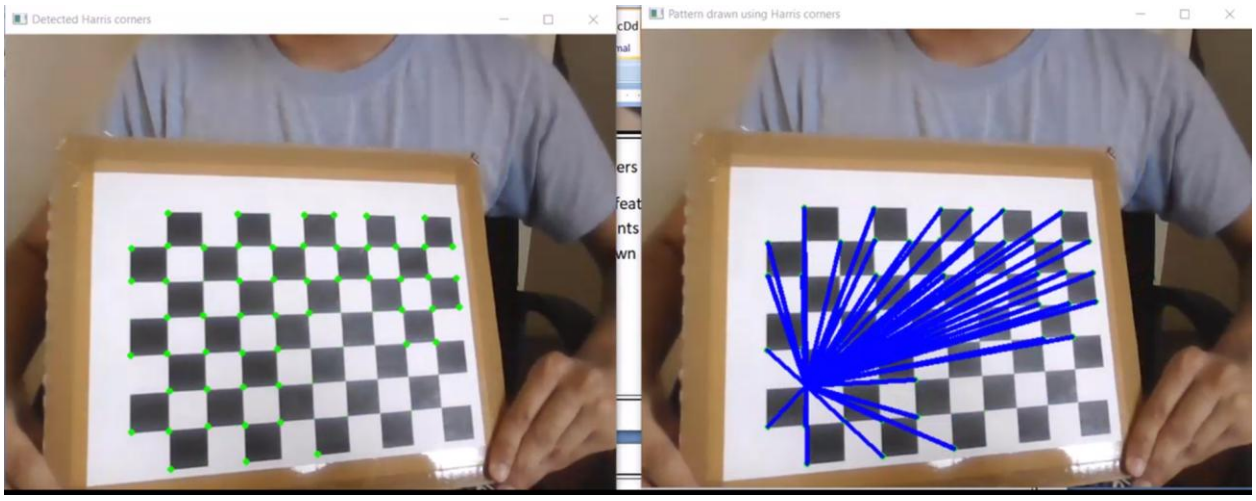


Harris corners detected on checkerboard

Here we see that the strongest features detected by the Harris corner detector. We can use these as feature points and augment or draw virtual shapes in a continuous video stream as shown in the video demo below:

Video demo:
https://drive.google.com/file/d/1NIGonu5cLbotaC7ELh3uTQJJHAfVyirj/view?usp=

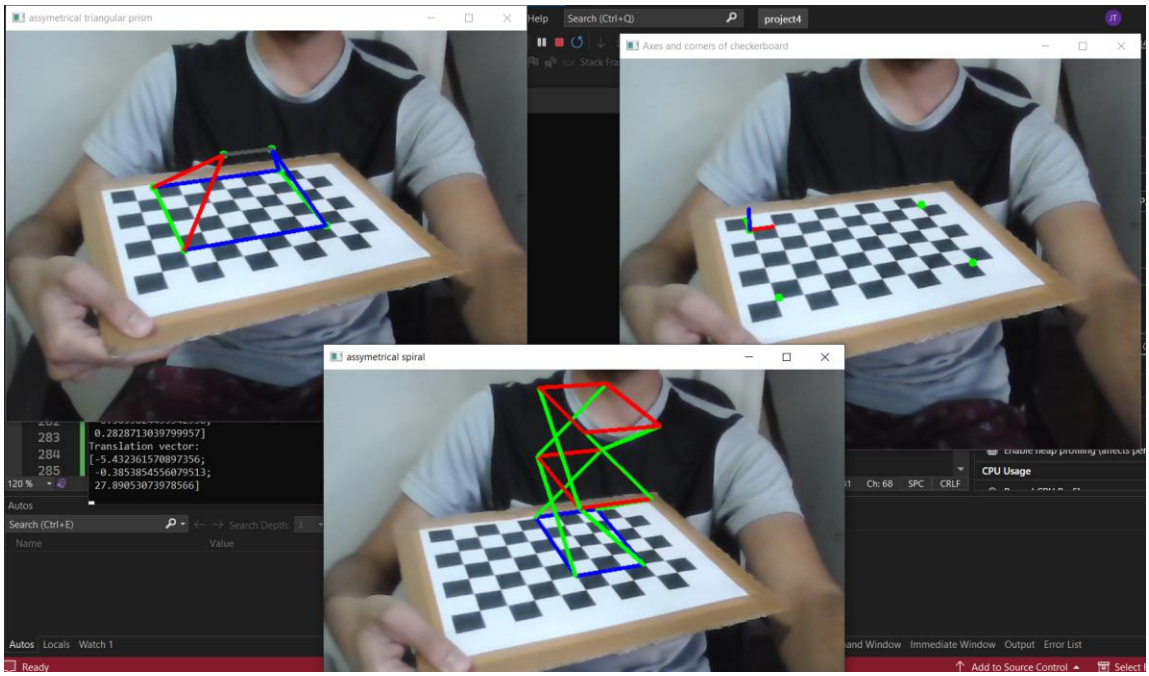
[share link]

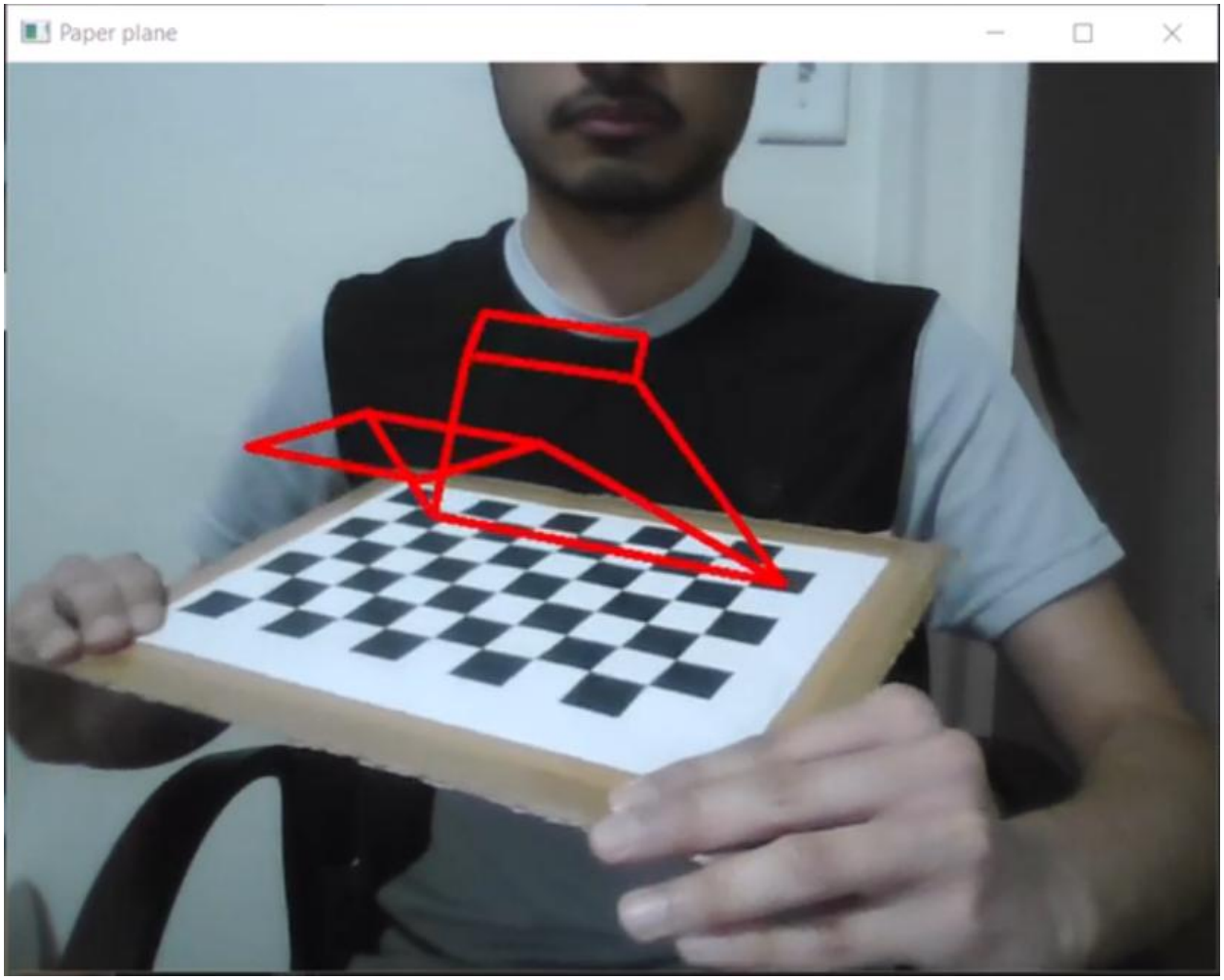


The above is just a random pattern formed using the Harris corners detected. To have a virtual object steadily projected onto a known pattern, there would have to algorithms that detected a particular arrangement of the pattern corners(using Euclidean distance between corners etc.) in the pixel plane(as outputted by the goodFeaturesToTrack() function).

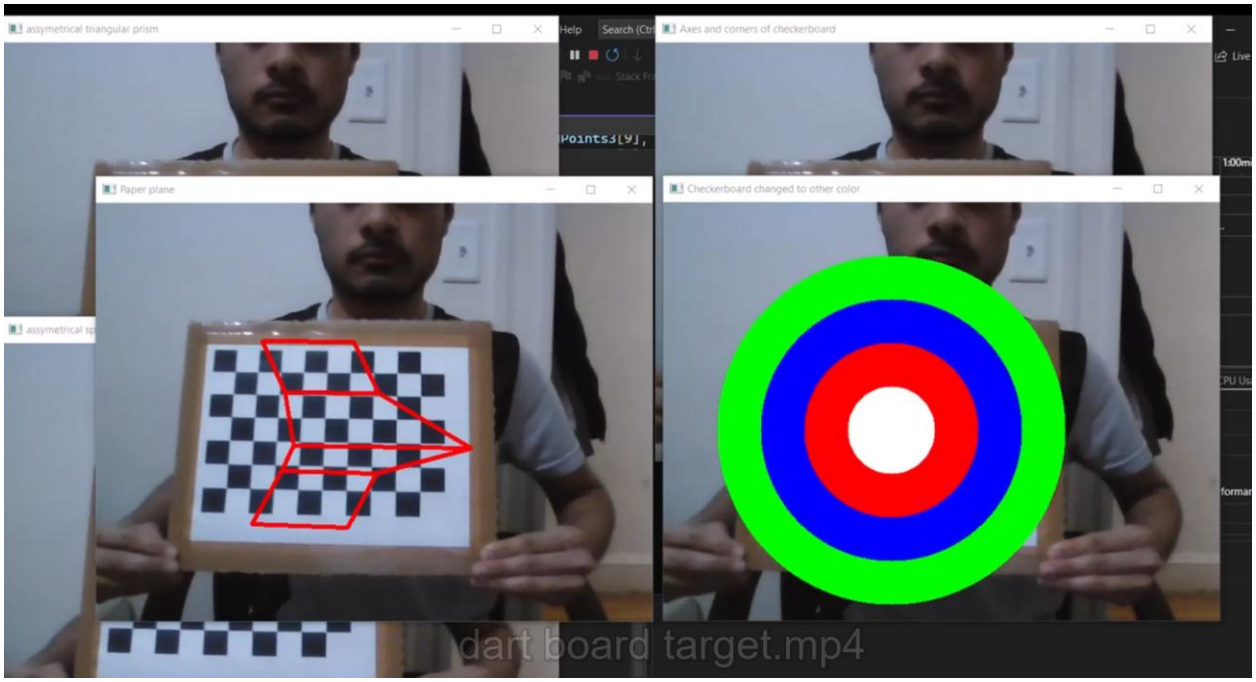
EXTENSIONS:

-More creative virtual objects:





Video demo of plane object:
https://drive.google.com/file/d/1hJWKmzMOHCeWOyPIJQs1etxXjlexXONg/view?usp=share_link



Video demo of target shape onto checkerboard(completely occluding the checkerboard): https://drive.google.com/file/d/1mS86xnOJ9sdSasP2PFDmtrLA0PmhEIX1/view?usp=share_link

REFLECTION:

The project successfully implemented a system to calibration a given camera and use the camera calibration parameters to augment a virtual 3D object onto a know checkerboard pattern. The implemented code tracks the pose of the checkerboard giving a real-time augmented reality system tracking and projecting the virtual object onto the checkerboard. Further, Harris corner features were calculated for object which could be used as feature points for augmenting 3D virtual objects onto.

ACKNOWLEDGEMENT AND REFERENCES:

- Computer Vision: Algorithms and Applications, 2nd ed. Links to an external site., Rick Szeliski
- Camera Calibration using Zhang's method(Cyrill Stachniss)
- Camera calibration and 3D reconstruction-OpenCV