CS5330_Project4

A short description of the overall project in your own words. (200 words or less)

This project focuses on camera calibration and object augmentation, and it aims to help us understand the steps like camera calibration, calculating camera positions, generating and putting virtual objects on the target image. Exploring and understanding different features for camera calibration is also expected.

The system is able to detect chessboard and ArUco targets, then it calibrates the camera. After camera calibration, it is able to calculate the position of the current camera and then put a virtual object on the target image. The object should move and orients given the motion of the camera.

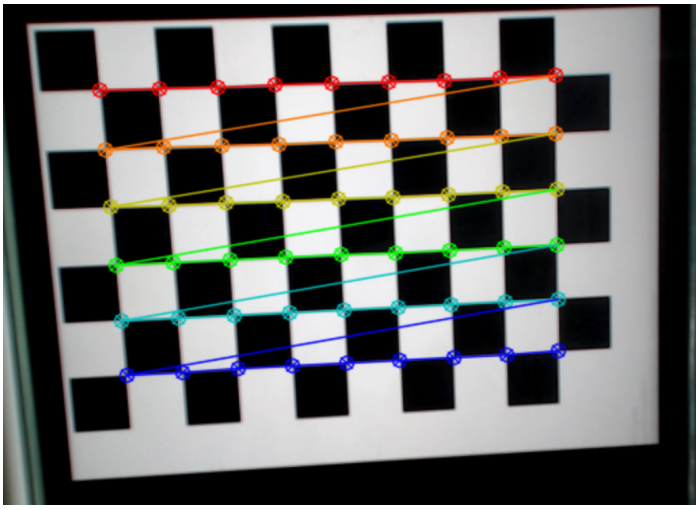
Demo Videos

- Demo of putting virtual objects on two targets at the same time https://youtu.be/St5iXxDZfCE
- Demo of overlay a picture on ArUco targets https://youtu.be/VtcGR1m9NBU

Any required images along with a short description of the meaning of the image.

Select Calibration Images

chessboard corners highlighted



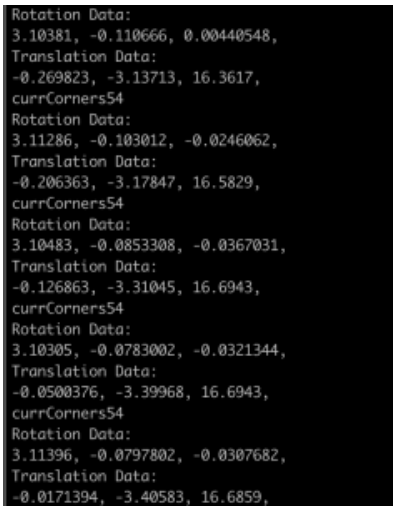
Calibrate Camera

Display the camera matrix, distortion coefficients, and re-projection error in the terminal

```
calibrate camera
Chessboard Camera Matrix:
685.062, 0, 479.843,
0, 679.721, 283.911,
0, 0, 1,
Chessboard Distortion Coefficients:
0.075033, -0.127571, 0.00259739, 0.000880873, -0.0324982,
Chessboard Re-projection Error: 0.213766
```

Calculate Current Position of the Camera

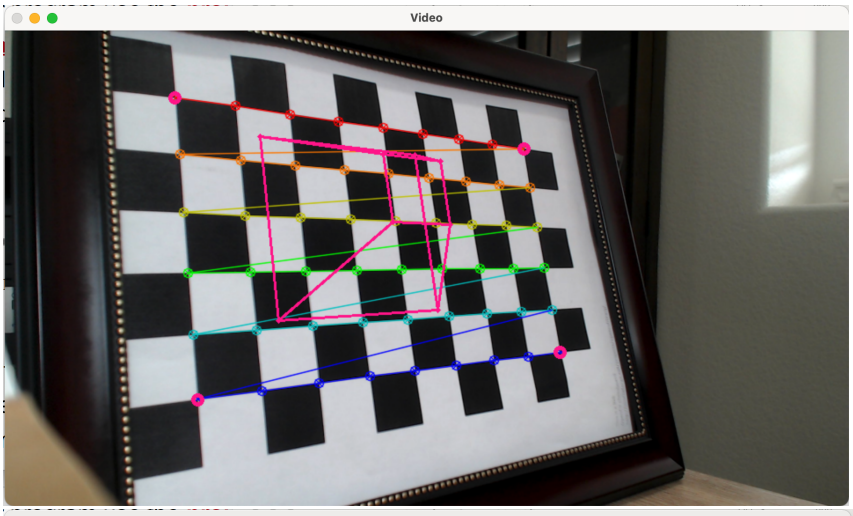
When testing, the system displays the rotation and translation data in real-time.

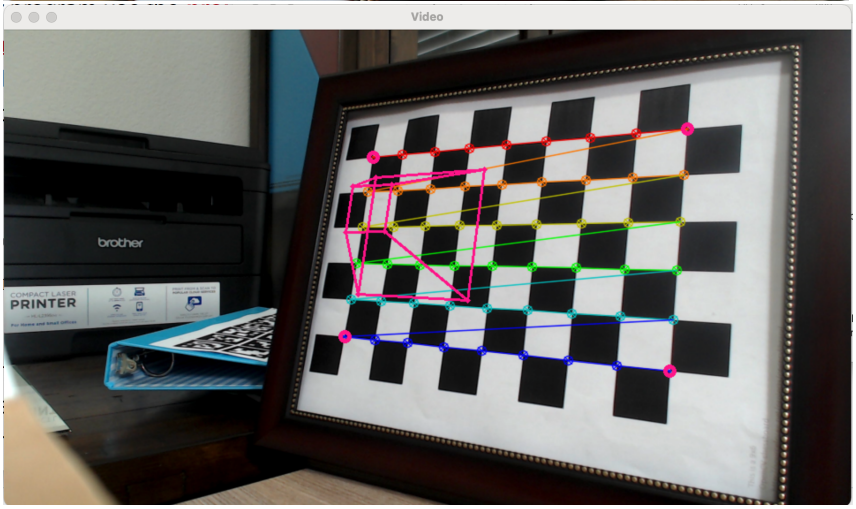


Project Outside Corners and Create a Virtual Object

The four outside corners are highlighted in pink, and the virtual object is placed on the chessboard.

Refer to the demo videos for system in action.



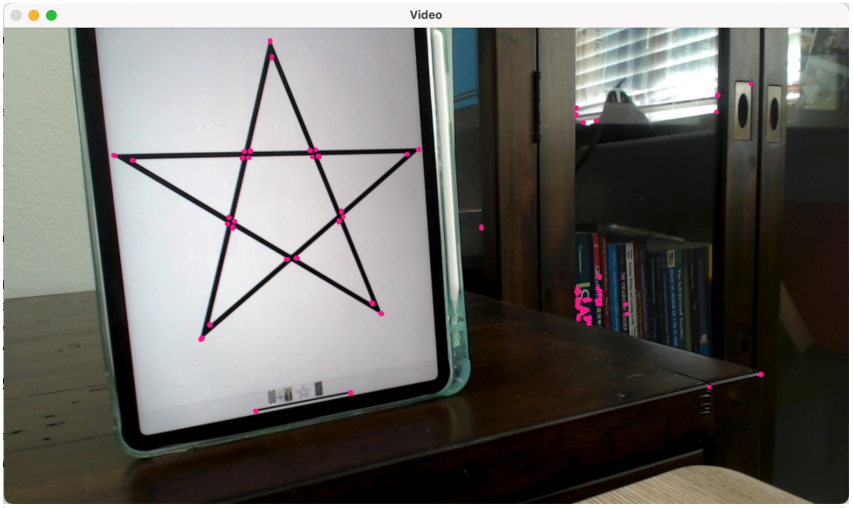


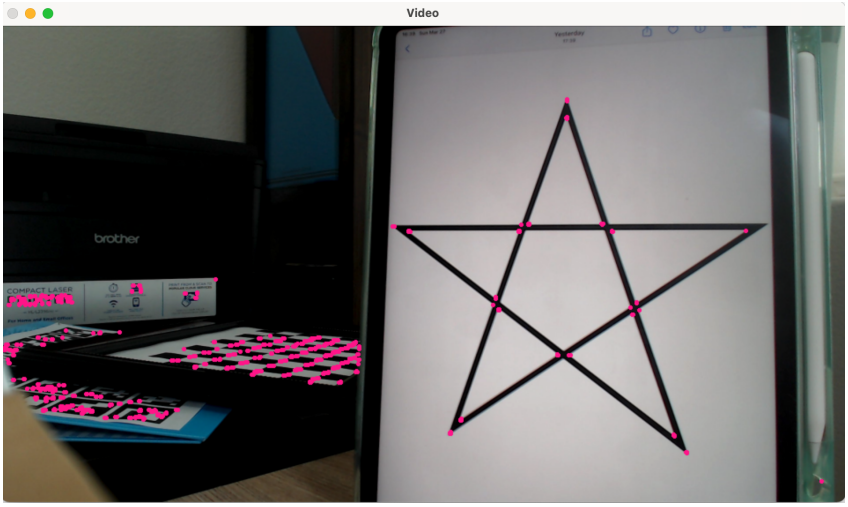
Detect Robust Features

Harris Corners Detector is implemented for this task. Images show that corners are detected in a picture of a star and the chessboard.

Similar to the chessboard corners, before using the Harris feature points as the basis for putting augmented reality, we first need to find the correlations between the 2D coordination in the image and the 3D coordination in the real world. Once we are able to project the points in the image to a 3D coordinate system, we can calibrate the camera and then get the camera position for further use.

Also, from the image, we can see that HarrisCorners can detect more corners(even the corners in a letter) so it may be able to be used for more detailed feature detection.



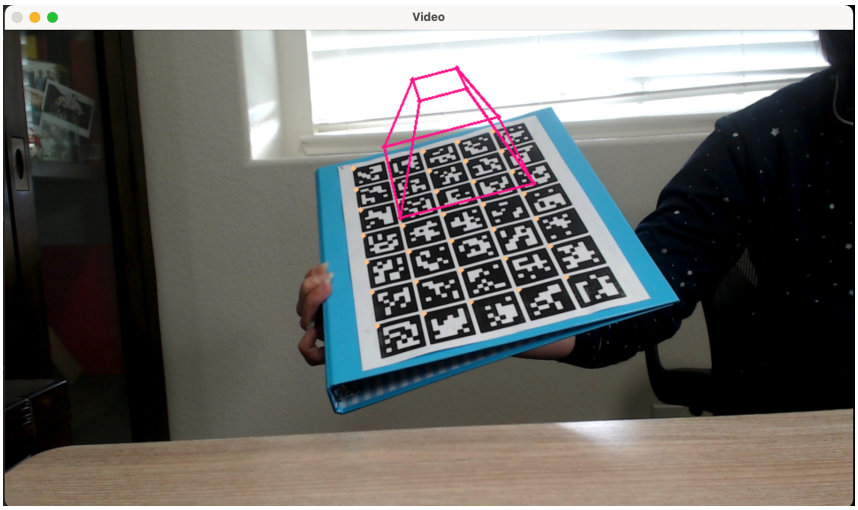


A description and example images of any extensions.

Extension 1 - ArUco

Besides chessboard, the system can also detect ArUco targets and then calibrate the camera using the features detected.

The image shows that the top left corner of each ArUco target is highlighted, and a virtual object is placed on the target.



The camera matrix, distortion coefficients, and re-projection error are also displayed in the terminal

```
calibrate camera
Aruco Camera Matrix:
683.624, 0, 496.277,
0, 686.828, 260.975,
0, 0, 1,
Aruco Distortion Coefficients:
0.154329, -0.619666, -0.00894774, 0.00945037, 1.06017,
Aruco Re-projection Error: 0.697774
```

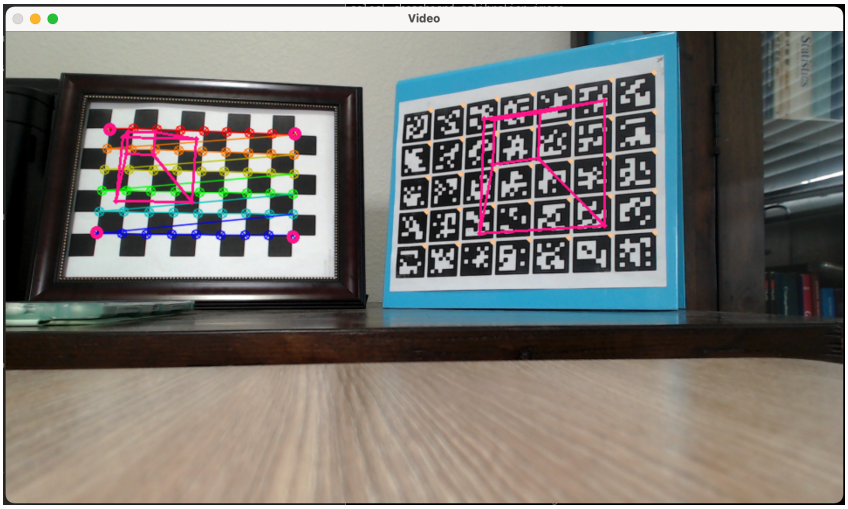
Extension 2 - modify the target

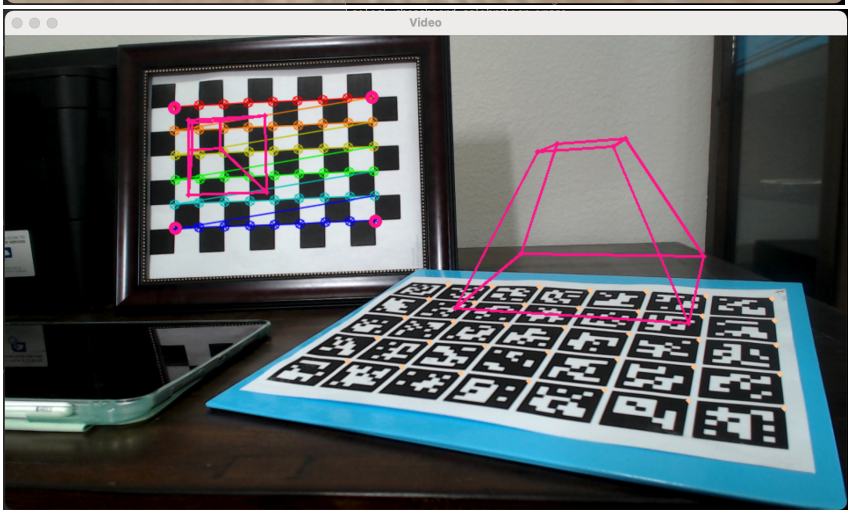
The system is able to overlay a picture on the ArUco targets.

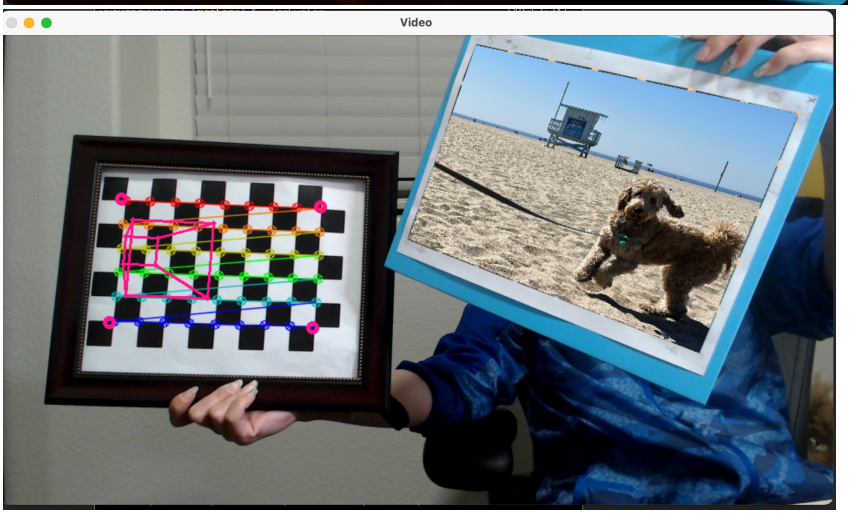
Use OpenCV findHomography() to find homography between the picture and ArUco target. In this step, the out four points of the target and picture are used. Then use warpPerspective() functions to warp together the picture and targets.

Extension 3 - Working with Multiple Targets at the Same Time

The system can detect features and put virtual objects or images on both the chessboard and ArUco targets at the same time







A short reflection of what you learned.

This is my first time working with camera calibration, and I learned about the process of camera calibration and the basic concepts like camera matrix and intrinsic parameters. How to create correlated coordinate systems for the image and real world is a challenge, and after this project, I get a better understanding of how the coordinates in the image and the ones in the real world are related to each other.

Besides, I also learned about augmented reality, where I can put simple virtual objects on the target image.

Acknowledgment of any materials or people you consulted for the assignment.

- Camera Calibration with OpenCV https://docs.opencv.org/4.x/d4/d94/tutorial_camera_calibration.html
- cornerSubPix https://docs.opencv.org/4.x/d2/d1d/samples_2cpp_2lkdemo_8cpp-example.html#a21
- Camera Calibration https://learnopencv.com/camera-calibration-using-opencv/
- ArUco https://www.uco.es/investiga/grupos/ava/node/69
- ArUco Marker Detection https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html
- Homography https://learnopencv.com/homography-examples-using-opencv-python-c/