Project 4 – Calibration and Augmented Reality

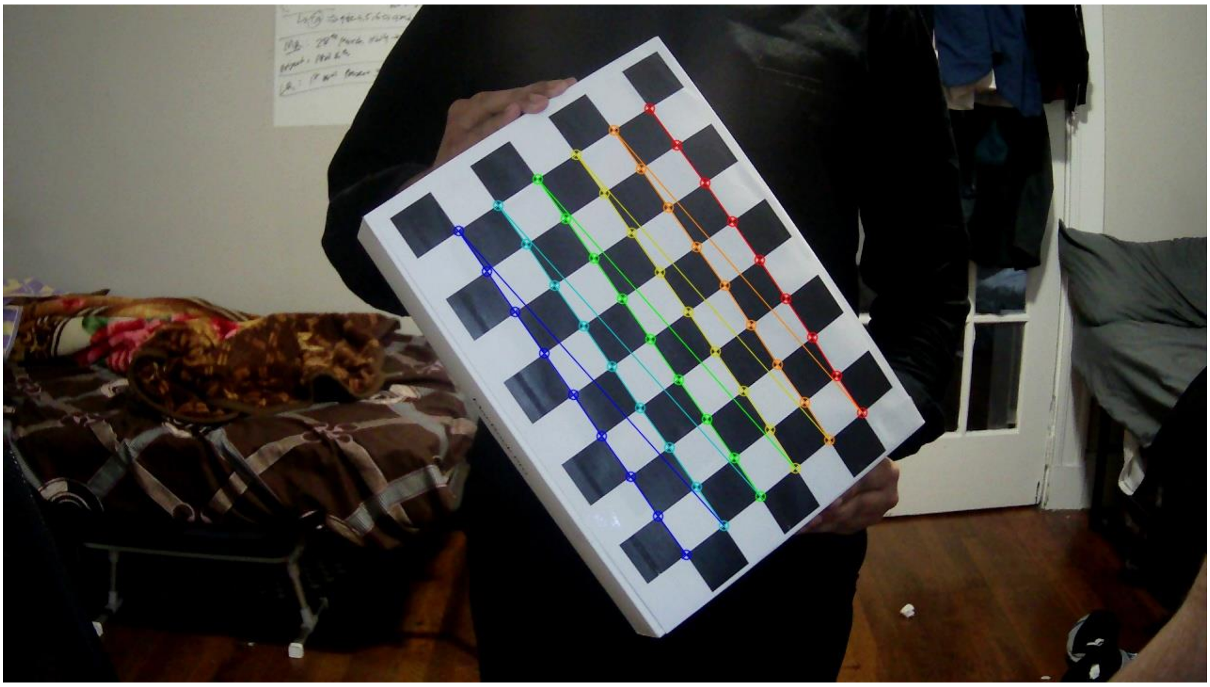
Abinav Anantharaman | NUID: 002774223

Satwik Shridhar Bhandiwad | NUID: 002920338

Introduction:

This project involves building a camera calibration system using OpenCV and Projecting Objects (AR). First we detect and extract the corners of a chessboard in an image or video stream. The next task is to allow the user to select calibration images and store the corner locations and corresponding 3D world points. Once enough calibration images have been selected, the user can calibrate the camera using the calibrateCamera function and generate the camera matrix and distortion coefficients. Finally, the program can use the calibration parameters to calculate the current position of the camera in real-time using the solvePnP function. The end result is a calibrated camera that can accurately determine its position and orientation in space. Finally, we project multiple objects on the Chessboard in real time.

Task2: Select Calibration Images



Calibration image with chessboard corners highlighted.

Task3: Calibrate the Camera

As part of this task, we have calibrated 3 cameras. They are External Webcam, HP Webcam and Asus laptop webcam. All of them have Reprojection error less than 0.5

For External Webcam the error is: **0.442412**

Task5: Project Outside Corners or 3D Axes

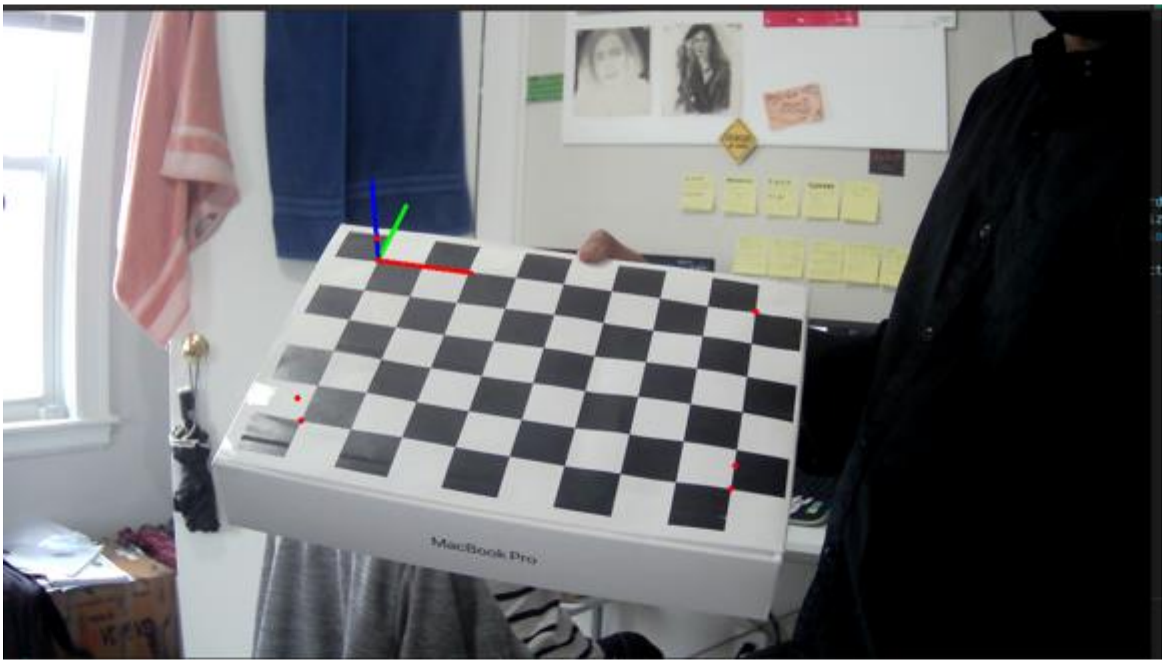


Image 1

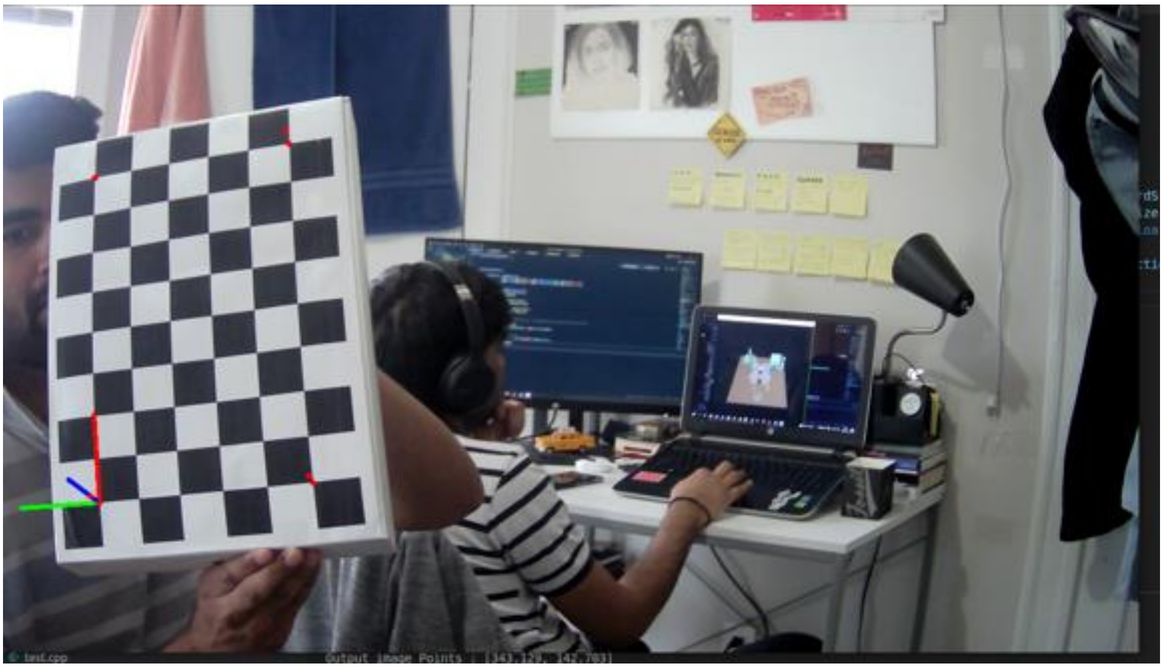
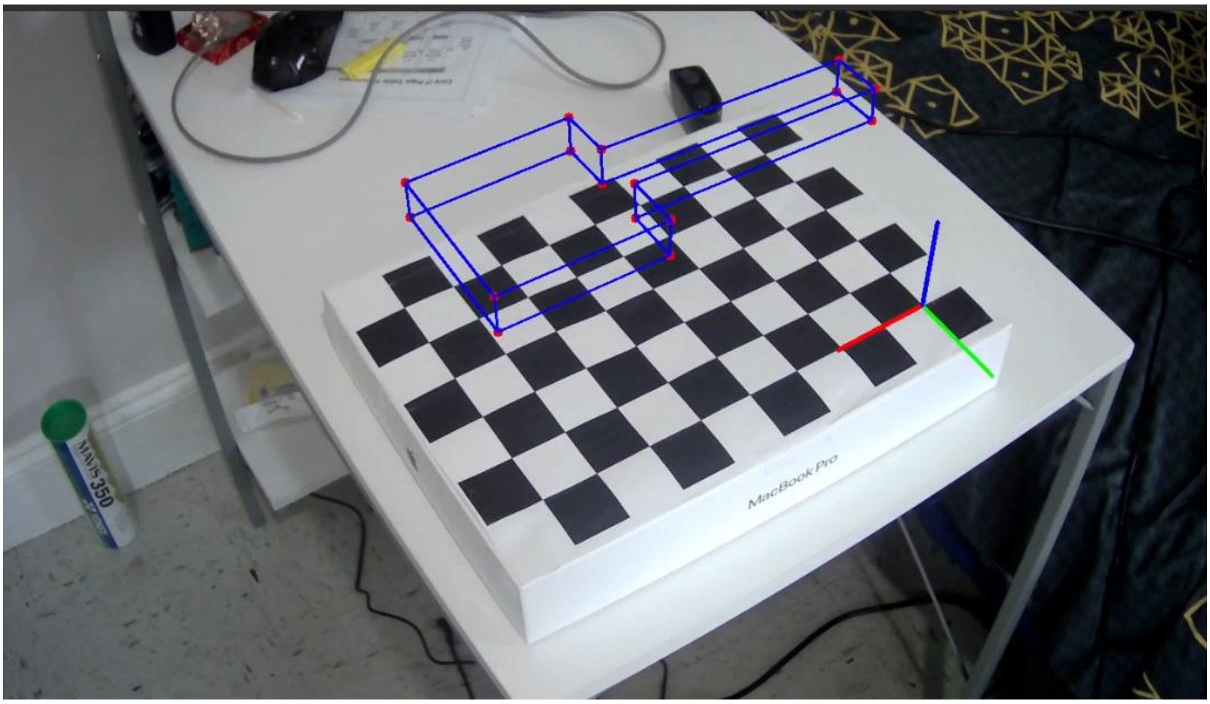
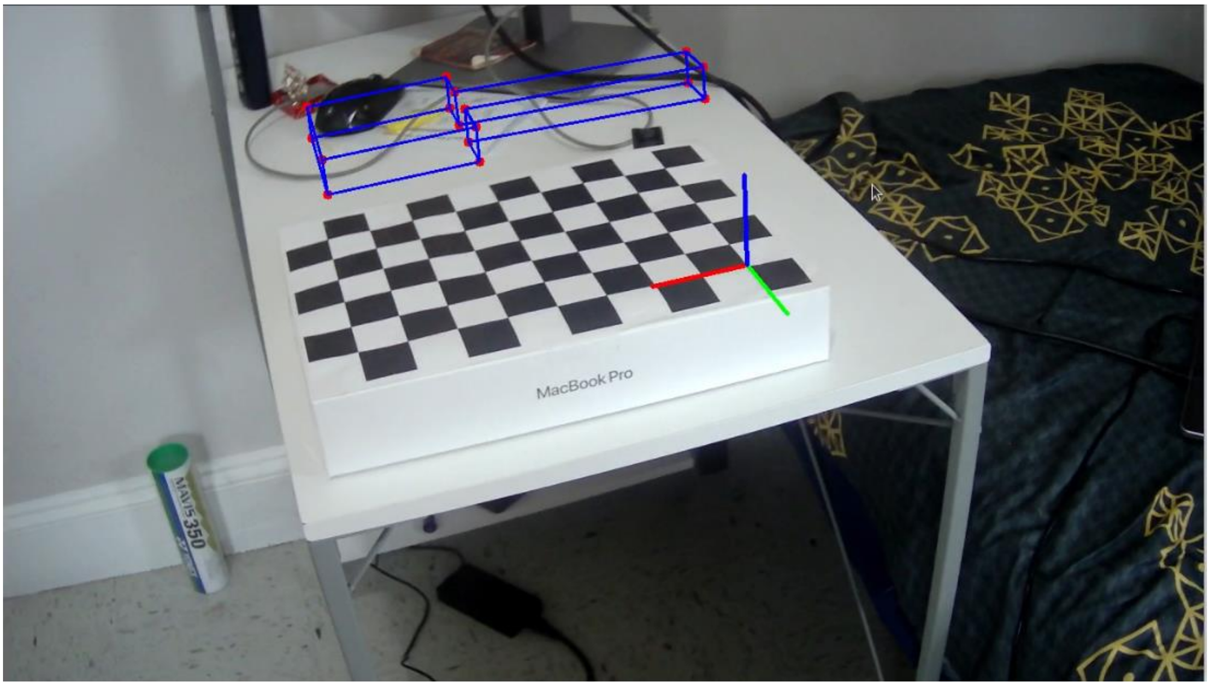


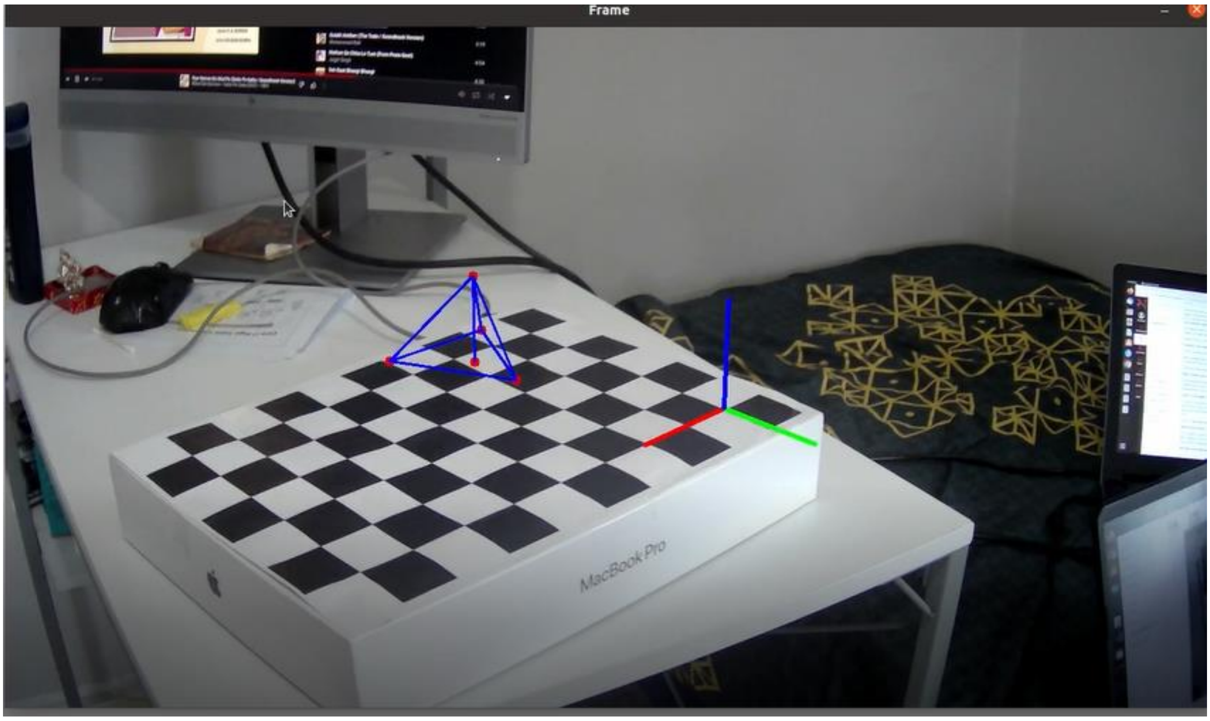
Image 1 and 2 projecting 3D Axes on the Chess board

Task6: Create a Virtual Object

Created a Spatula and Pyramid and projected it on the Chess Board.



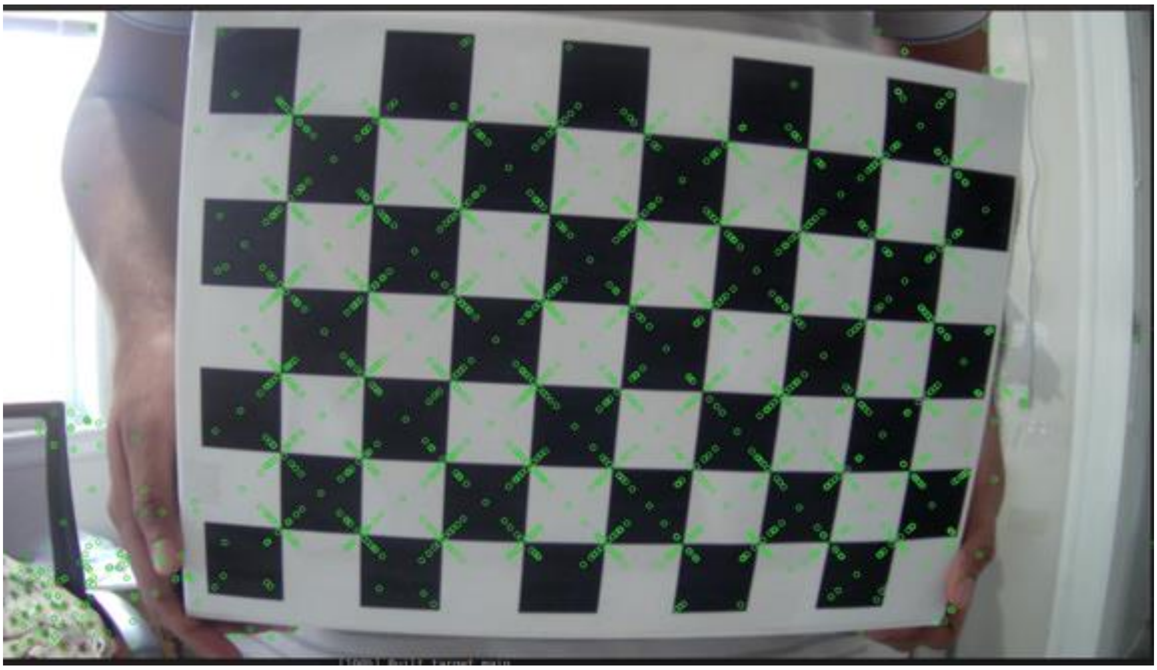


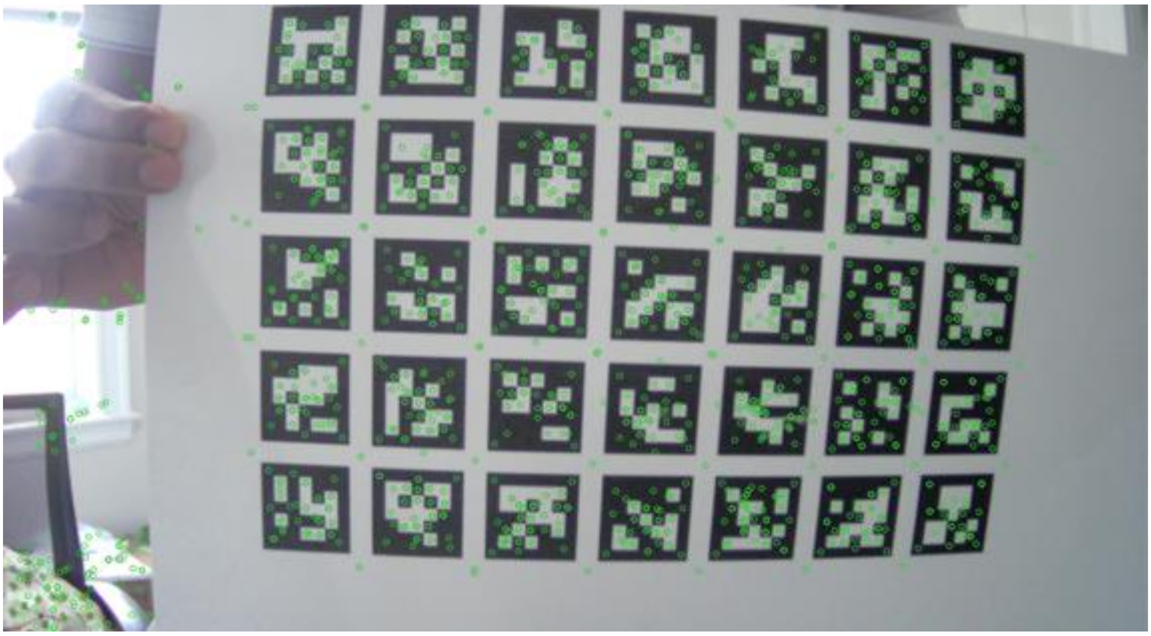


Demo Video of Spatula Projection:
https://drive.google.com/file/d/193T4PRdGKSXNuPLpKT7fI6Li7b7rd8F-/view?usp=sharing

Demo Video of Pyramid Projection:
https://drive.google.com/file/d/16no0FP5CnOTeX7HmwpyyuLSw0BBpDjuF/view?usp=share_link

Task7: Detect Robust Features - SURF



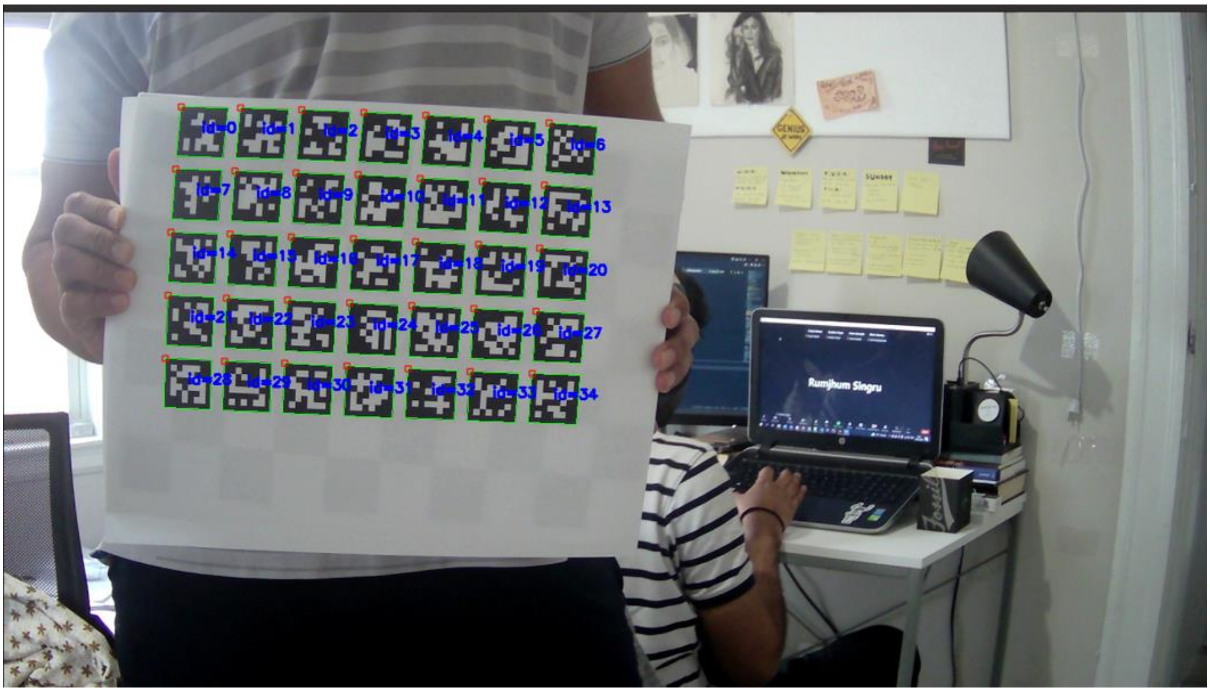


SURF feature points on Chessboard and ArUco markers.

Using SURF feature points for augmented reality:

1. The first step is to extract SURF features from the reference image, which will be used as a marker or a reference for the augmented reality. SURF is an algorithm for extracting robust and invariant features from an image.
2. After extracting the SURF features from the reference image, you need to store these features in a database, which will be used to match with the features extracted from the camera feed.
3. Next, you need to capture the camera feed in real-time using the camera of the device, which will be used to detect the reference image and overlay the augmented reality content on it.
4. Once you have captured the camera feed, you need to extract SURF features from it. These features will be used to match with the SURF features of the reference image stored in the database.
5. Now, you need to match the SURF features extracted from the camera feed with the SURF features stored in the database. You can use a feature-matching algorithm, such as the FLANN (Fast Library for Approximate Nearest Neighbors) algorithm, to find the best matches.
6. Once you have found the best matches, you need to estimate the homography between the reference image and the camera feed. Homography is a transformation matrix that maps the points in one image to the corresponding points in another image.
7. Finally, you can use the estimated homography to overlay the augmented reality content on the reference image in real-time. You can use OpenGL or other graphics libraries to render the augmented reality content.

Extension1: ArUco Marker Integration



Integrated ArUco module in OpenCV to generate ArUco markers and detect them and display the IDs.

Extension2: Multiple Camera Calibration

Below are the reprojection errors of 3 different cameras.

External Webcam: 0.442412 Resolution: 1280 * 720

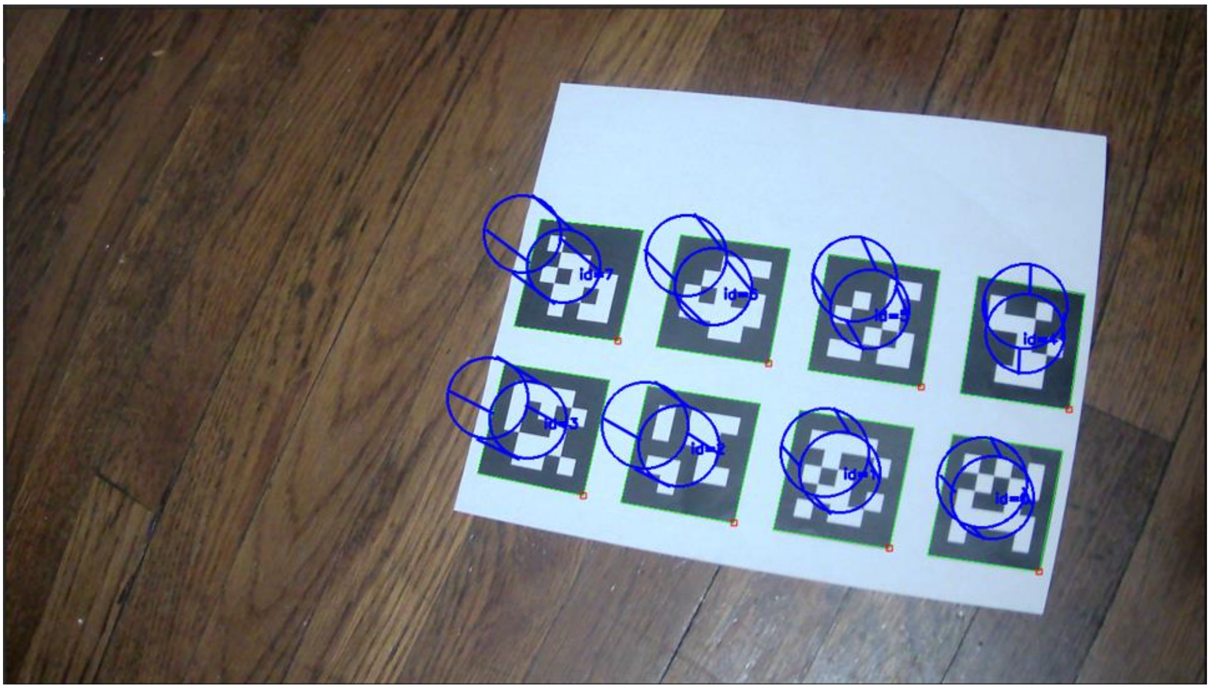
HP Webcam: 0.262084 Resolution: 640 * 480

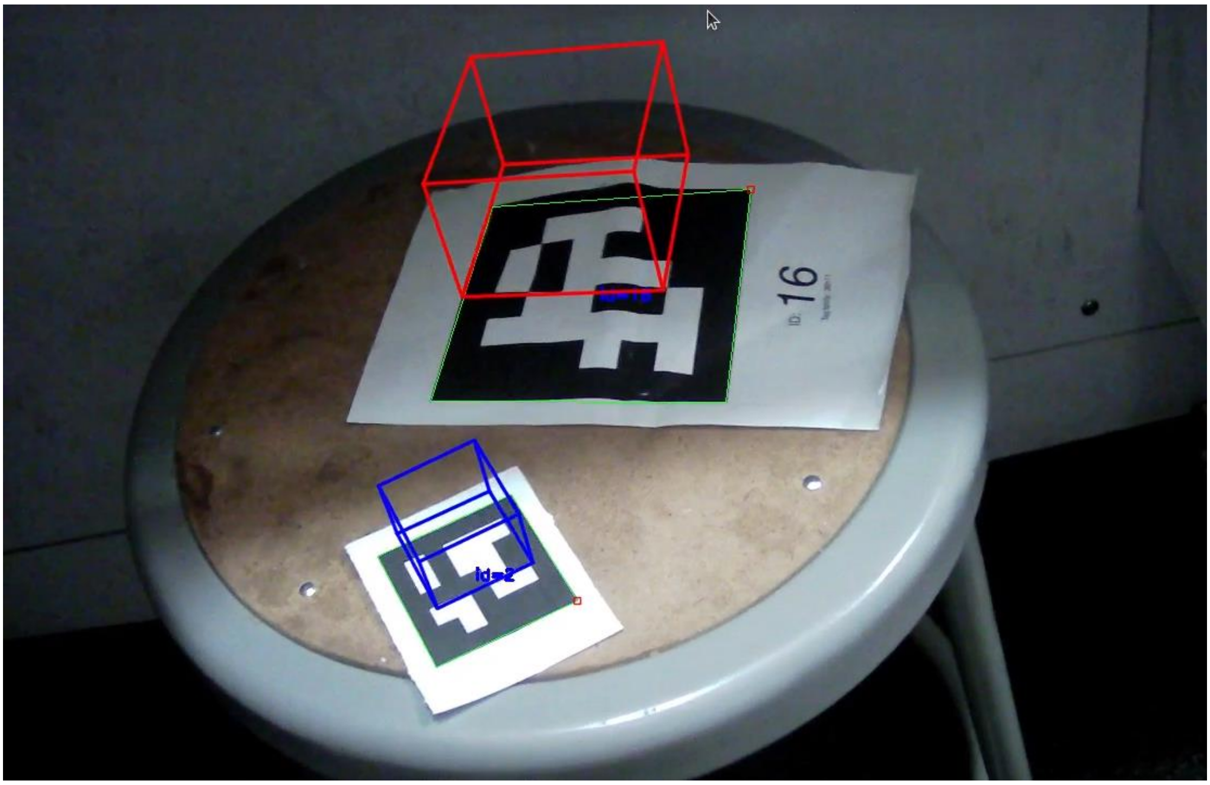
Asus Webcam: 0.388337 Resolution: 1280 * 720

All the three cameras have error less than 0.5 as mentioned in the project guidelines.

The HP Webcam with lower resolution had the least error in all the 3 cameras.

Extension3: Projecting Cylinders and Cube on Multiple ArUco markers





Demo Video: https://drive.google.com/file/d/1upn3L4-0m7jxZmYlDzYtqo6unGu3f0Rt/view?usp=share_link

Learning Outcomes:

1. This project provides a hands-on experience of working on camera calibration.
2. The project involves several stages, from detecting chessboard corners to calculating the camera position. It requires critical thinking to determine the best approach to tackle each problem and solve it effectively.
3. The project provides an opportunity to work on Augmented Reality.
4. This project provides a practical application of computer vision techniques. The skills and knowledge gained from this project can be applied to various computer vision applications, such as robotics and autonomous vehicles.

Acknowledgements:

George Lecakes Youtube channel: Part 14 to 19 (388) OpenCV Basics - YouTube

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

https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html