Project 4: Calibration and Augmented Reality

Hussain Kanchwala **Abdulaziz Suria**

Summary:

The project implements a real-time AR which generates virtual objects in image frame. The project encompasses calculation of intrinsic and extrinsic parameters of the camera with the help of the checkerboard pattern. The intrinsic and extrinsic matrices help to transform 3D world coordinates to 2D pixel coordinates which is useful in creating virtual objects in the image frame. We have followed the process outlined in the assignment and they are listed out below:

1. Detect and extract the corners.
2. Select Calibration Images and Fixing the world frame.
3. Camera Calibration
4. Augmented Reality
5. Creating Virtual Objects
6. Detect Robust Features.
7. Extension

We have provided the required images and required insights further ahead in the report.

NOTE: Checkerboard pattern is used for calibration

1. Detect and extract the corners

The corners of the checkerboard were detected utilizing the opencv functions “findChessboardCorners”, “cornerSubPix” and “drawChessboardCorners”.

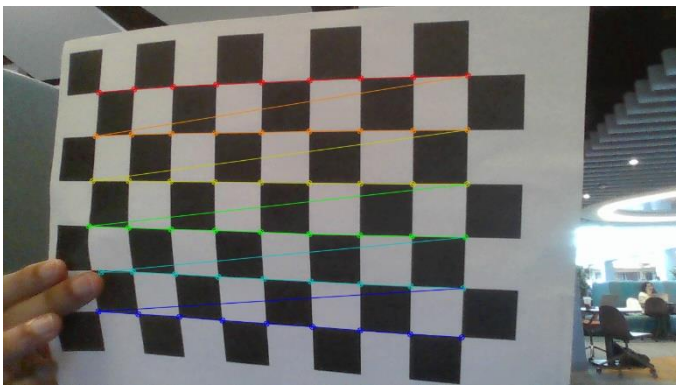

Figure 1: - Checkerboard Corner

The location of the first corner that is top left is (182,163) and the total number of corners found are 54.

Limitations and Challenges
If the checkerboard image is too far away then our system is not able to detect it at times. The lighting of the surroundings should be stable as well or else it sometimes fails to recognize if a checkerboard is present or not. Sometimes, the system may misbehave but that occurs rarely.

2. Select Calibration Images and Fixing World Coordinate Frames: -

The checkerboard pattern is placed in front of the camera in different orientations and on the click of key ‘s’ the corresponding corners are recorded, and the images are saved in the folder specified.

In order to calibrate the camera, we need to fix a world coordinate frame. For the ease I have fixed my world coordinate frame on the first corner of the checkerboard in real life with it’s x along the direction of the columns, z pointing towards the camera and y given by the middle finger of my left hand stretched out perpendicular to the thumb (z-axis) and index finger (x-axis).

Here I don’t care about the actual size of the object in real world. So, after defining the world frame I assumed that the (0,0,0) is my first corner in real life followed by (1,0,0) and so on.

Please note that according to my definition of the world coordinate frame, to go to the next row underneath the first one you must traverse along the -y axis.

3. Camera Calibration

After collecting a minimum of 5 checkerboard corners in image frame and their corresponding location in world coordinate frame the “calibrateCamera” function from opencv is utilized to calculate the intrinsic matrix.

On the click of key ‘c’ the calibration starts and registers the intrinsic matrix in a .yml file.

Reprojection Error: - 0.178742

Camera Matrix

$$\begin{matrix} 1 & 0 & 640 \\ 0 & 1 & 360 \\ 0 & 0 & 1 \end{matrix}$$

Before

$$\begin{matrix} 920.347 & 0 & 652.184 \\ 0 & 920.347 & 365.316 \\ 0 & 0 & 1 \end{matrix}$$

After

Distortion Coefficients

[0,0,0,0,0,0,0,0] --- before [-0.22, 0.90,-0.0032,0.0008,-1.326] --- After

4. Augmented Reality

Once the intrinsic matrix is estimated. Another essential component remaining is the extrinsic matrix which in combination with intrinsic matrix will help transform objects in 3D world frame to 2D image frame.

In order to get the extrinsic matrix “solvePnP” function of Opencv is utilized.

To check if the transformations are correct. I tried to move my checkerboard along the positive direction of the x axis and observed the translation.

Frame 1: - [5.193583858621791; -0.6966614494592639; 29.45999500911332]
Frame 2: - [5.853528379929856; -0.3820118147796506; 29.54673314640872]
Frame 3: - [6.368789009389961; -0.1920180643973775; 29.65167915805388]
Frame 4: - [7.050770073503063; -0.1260470546155163; 30.45022155015253]

Observing the above by moving the checkerboard along the x axis the translations look correct.

The extrinsic matrix and the intrinsic matrix are defined properly. The “projectPoints” function is used to transform the points from 3D world coordinates to 2D world coordinates.

Again, to check if the transformations are correct, I defined points in 3D world coordinate and observed its projection on 2D image frame and it’s correct which confirms are transformation calculations.

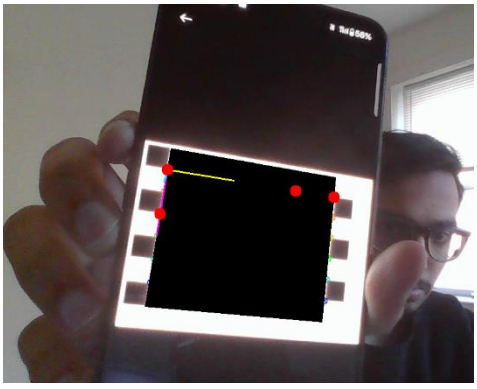

Figure 2: - Projection

The points defined in 3D world coordinate are: -

[0,0,0], [8,0,0], [6,0,0], [0, -2,0]

5. Creating Virtual Object

We have displayed a Triangular Prism as our target virtual object. It is a combination of 2 triangles connected with 3 rectangular surfaces. We have defined certain variables such as x,y length and height to adjust the triangle if needed.

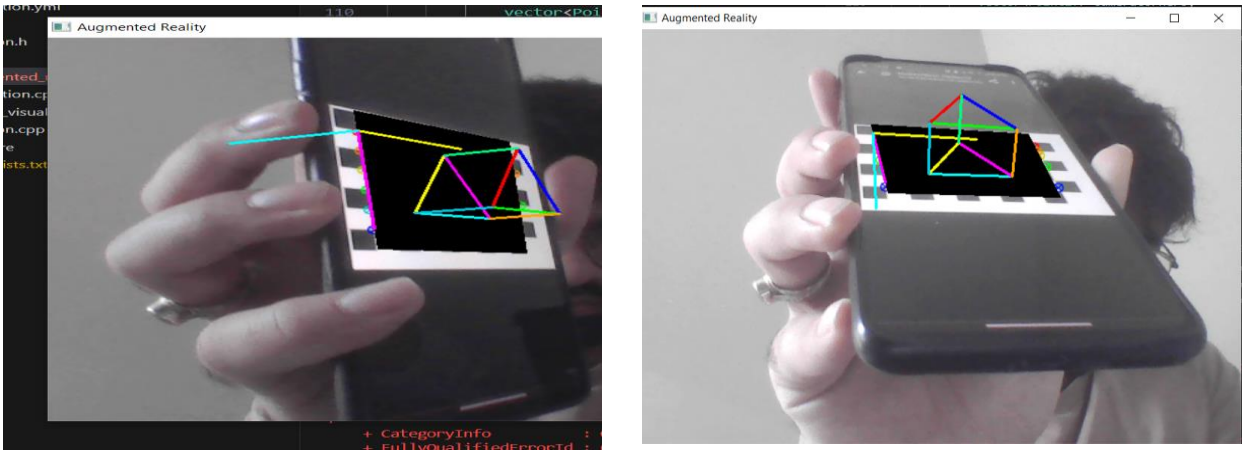

Figure 3: - Prism

6. Detecting Robust Features

Shi-Tomasi Corner Detector is utilized to find the corner features in an image frame. The scoring function of Shi-Tomasi is R = min(L1,L2). So, if the R value is greater than a given threshold only then the pixel is considered as a corner.

- Only when L1 and L2 are above a minimum threshold value it is considered as a corner.
- When either of L1 or L2 is below the minimum threshold it is considered as a edge.
- When both are below the threshold then it is considered as a flat region.

experiments were done with different quality levels, corner values, minimum Euclidean distance between points, etc.

Utilization of such features for AR:

For simple objects, we can use Shi-Tomasi Corner Detection to find the corners of the object and estimate a pose and orientation of it based on the corner of the image. We can further perform feature matching using SURF feature vectors to find a known object and then use this information for projection.

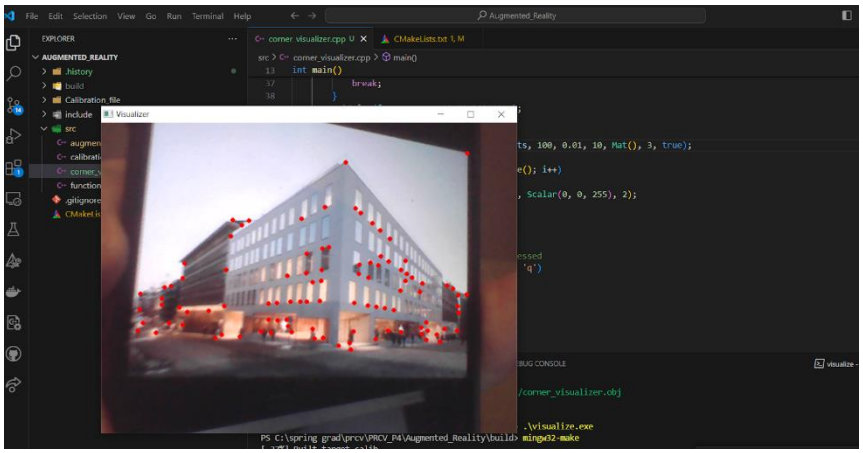

Figure 4: - Corner Features

EXTENSION (Homography Between Consecutive Images):

Implemented a custom Harris corner feature detector from scratch to find the corner in the scene, feature description is extracted for each corner from a patch size of 30*30 around the corner. Flann base feature matcher is used to return the matches. RANSAC

Figure 5: - Images for Homography

is used to reject outliers and give the appropriate transformations between the images.

Homography Matrix (Image 2 wrt Image 1)

[-0.1204777163987411, -0.008200761108820951, 40.26479024921048.

-0.8191777233865015, -0.06237251891155116, 309.7202110248207.

-0.002677335152476304, -0.0001528416548565282, 1]

The application of this procedure is in 3D Reconstruction and Panorama.

Reflection:

After completing this project, we learned about the camera calibration using checkerboard pattern and implementing virtual reality. In camera calibration we learned to calculate intrinsic and extrinsic parameters which can then be used to project 3D world points on the image plane. In Augmented Reality, we experimented with projecting a prism in the image frame. As an extension I learned in detail regarding feature extraction, feature matching and find homography between the two frames.

Acknowledgements:

Thankful for the support from the TAs who helped in clearing our doubts during the Office Hours.

- https://docs.opencv.org/4.x/d4/d94/tutorial_camera_calibration.html
- https://docs.opencv.org/4.x/dd/d1a/group__imgproc__feature.html#ga1d6bb77486c8f92d79c8793ad995d541
- OpenCV documentation