Assignment 3

Camera calibration and Augmented Reality

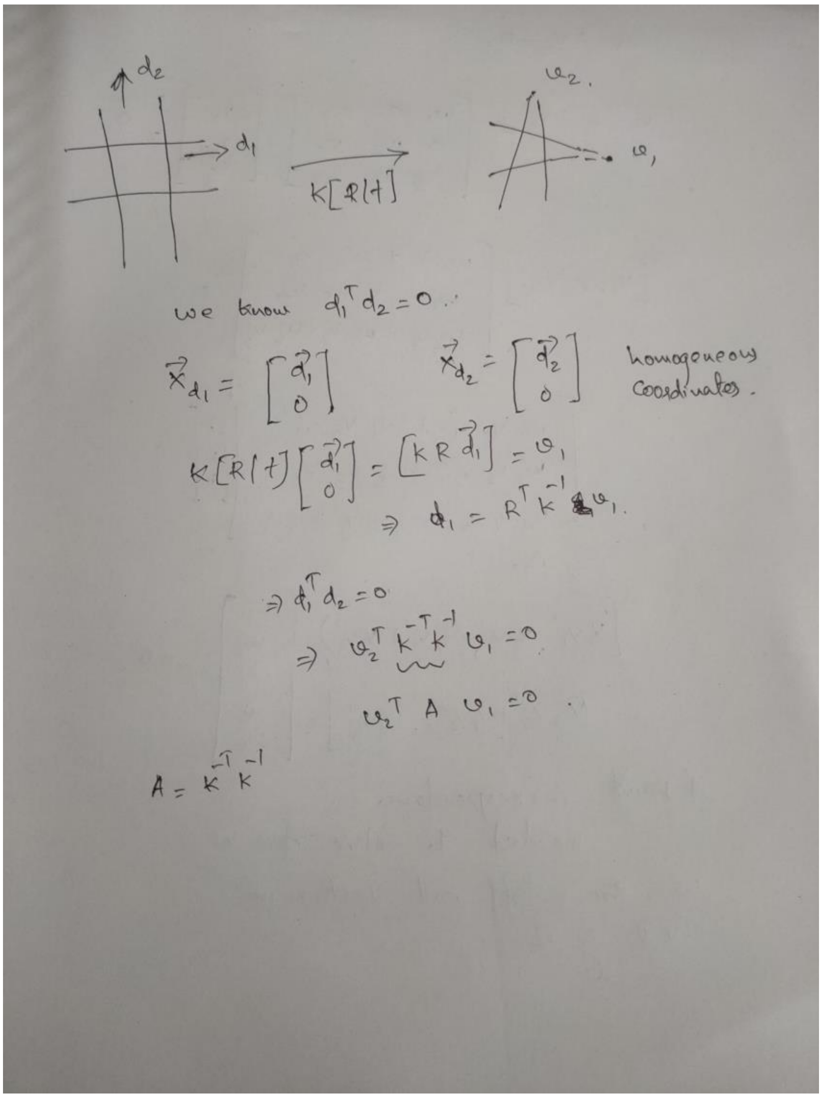
Hari Krishnan CK

2021EEY7583

Part 1: Camera Calibration

Camera calibration is done using the Vanishing point method. The idea is to find the parallel lines in the real world in the image and hence find the vanishing points. Once we have a set of vanishing points we can solve the equation Aw=0 to find the K matrix by cholskey decomposition.

Theory



The above derivation reaches the point that $v_2^T A v_1 = 0$ where v_1 and v_2 are the vanishing points of a set of parallel lines.

$$v_1'^T A v_2 = 0.$$

$$[v_1' \;\; v_2' \;\; v_3'] \begin{bmatrix} w_1 & 0 & w_4 \\ 0 & w_2 & w_5 \\ w_4 & w_5 & w_6 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = 0.$$

$$[v_1' \;\; v_2' \;\; v_3'] \begin{bmatrix} w_1 v_1 + w_4 v_3 \\ w_2 v_2 + w_5 v_3 \\ w_4 v_1 + w_5 v_2 + w_6 v_3 \end{bmatrix} = 0$$

$$\Rightarrow \begin{bmatrix} w_1 v_1 v_1' + w_4 v_3 v_1' \\ w_2 v_2 v_2' + w_5 v_3 v_2' \\ w_4 v_1 v_3' + w_5 v_2 v_3' + w_6 v_3 v_3' \end{bmatrix}^T \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = 0.$$

$$\Rightarrow \overbrace{\left[v_1 v_1', \; v_2 v_2', \; (v_3 v_1' + v_1 v_3'), \; (v_3 v_2' + v_2 v_3'), \; v_3 v_3' \right]}^{\Omega} \begin{bmatrix} w_1 \\ w_2 \\ w_4 \\ w_5 \\ w_6 \end{bmatrix} = 0.$$

4 point correspondance needed to solve for w

w is the left null vector in SVD of Ω

Once we have four sets of vanishing points, the solution can be obtained and if we do Cholesky decomposition of A we obtain the K matrix.

Algorithm :

Assumption : s=0 and fx=fy

Step 1: Identify at least three sets of parallel lines in the real world assuming $f_x = f_y$ and scale s=0.(If we do not assume $f_x = f_y$ then we need 4 sets of parallel lines). Make sure that these vanishing points do not lie in the same plane.

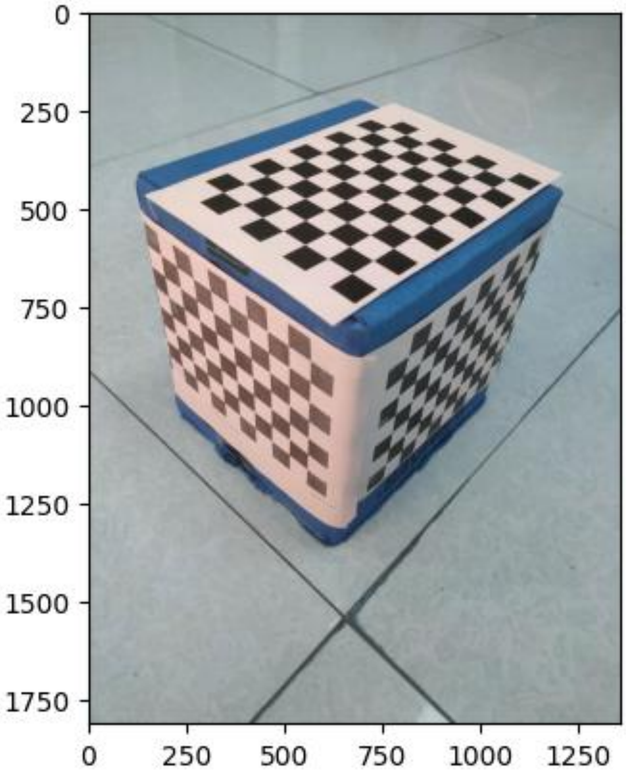


Figure 1: A checkerboard cube in which parallel lines were found.

Step 2: Find the vanishing points of these parallel lines in the image by finding the intersection of these lines.

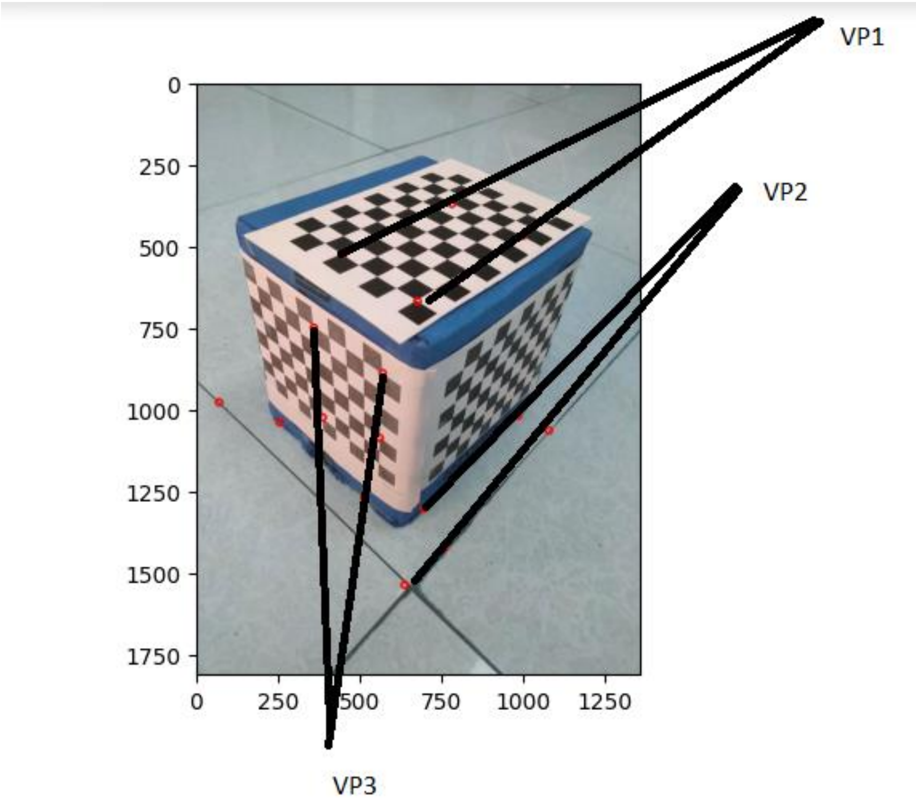


Figure 2: Figure showing the Vanishing points(VP1,VP2,VP3)

Step 3: Do SVD of A where A= array([[x1, 0., x3],

[0., x2, x4],

[x3, x4, x5]])

Step 4: Do Cholesky decomposition of A to get K^{-T} and finally get K by taking the inverse of transpose of K^{-T} .

Result: The camera calibration matrix obtained for the camera was

```
The intrinsic matrix obtained is  [[1.21489001e+03 0.00000000e+00 4.1691696
0e+02]
 [0.00000000e+00 1.21489001e+03 8.81331195e+02]
 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]
```

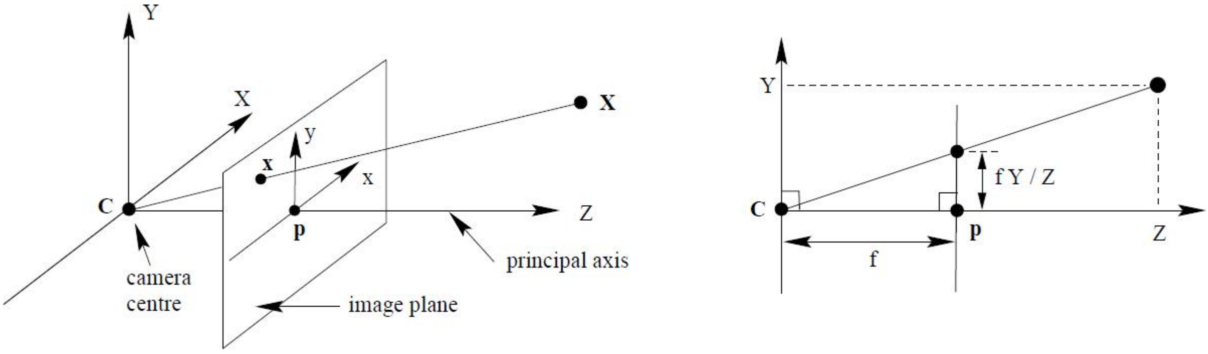
Part 2: Augmented Reality.

In this part we try to render a 3D model to a video taken with the same camera as above. The steps involved in doing it is described below.

Theory:

Assumption:

We are assuming pinhole camera model which is given below



A reference image of the scene where we want to augment the 3D model is given. Assume that the Z coordinate of the plane is 0.

Below is the calculation for finding the rotation matrix R and translation matrix t. (x,y,1) is the image coordinate and (X,Y,Z,1) is the real world coordinate.

$$\begin{bmatrix} sx \\ sy \\ s \end{bmatrix} = k[R|t]\begin{bmatrix} x \\ y \\ 0 \\ 1 \end{bmatrix}$$

$$\Rightarrow k\,[R_1\ R_2\ R_3\ t]\begin{bmatrix} x \\ y \\ 0 \\ 1 \end{bmatrix}$$

$$= k[R_1\ R_2\ t]\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$= H\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$H = K\,[R_1\ R_2\ t]$$

$$[R_1 R_2\ t] = K^{-1}H$$

Once we have R_1, R_2 and t we have to first orthonormalize the rotation matrix using Gram Schmidt’s process using the following method

Given R_1, R_2

$$u_1 = R_1 / \|R_1\|$$

$$u_2 = R_2 / \|R_2\|$$

$$u_2 = u_2 - \langle u_1 . u_2 \rangle u_1$$

$$u_2 = u_2 / \|u_2\|$$

return u_1, u_2

Algorithm:

Given is the intrinsic matrix K obtained from Part 1, the reference image and video where we want to render the 3D model.

Step 1: Take the image of a reference image which is the image which will be used to find the homography with the target images.

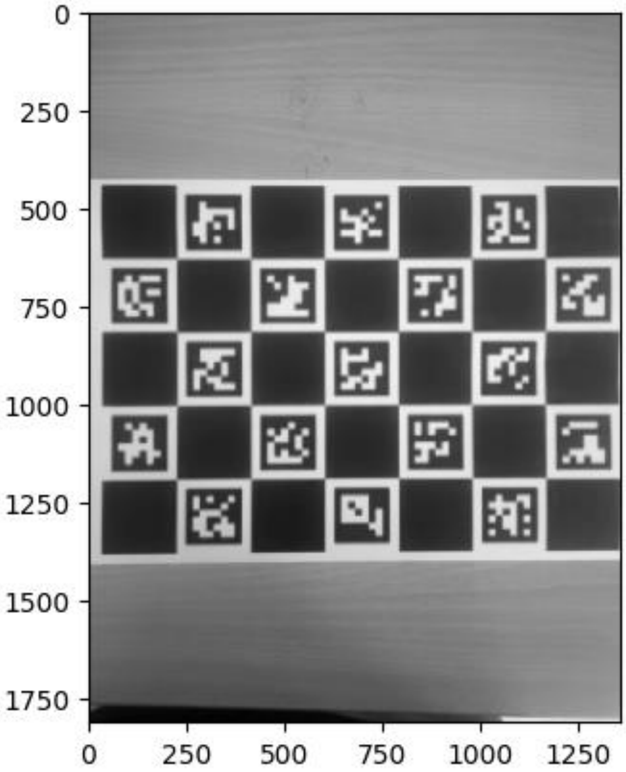


Figure 3: reference image

Step 2: Extract the frames from the given video using cv2.videoCapture.

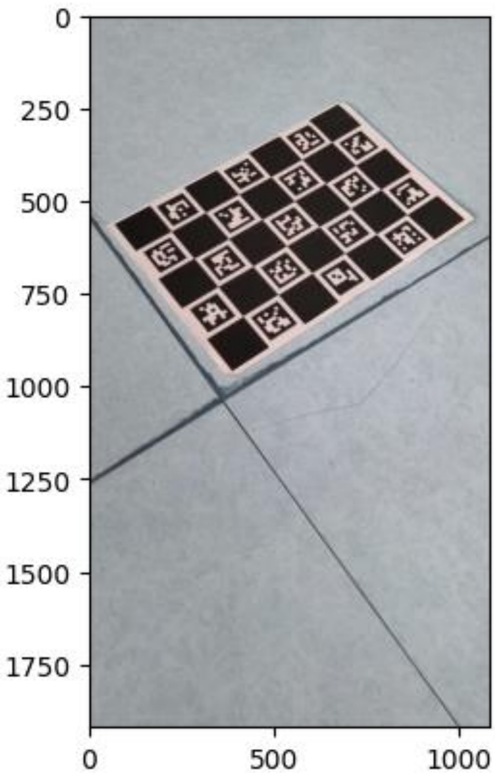


Figure 3: Image obtained from the video capture.

Step 3: Detect the interest points or corners using SIFT in Opencv or any other corner detection algorithm.

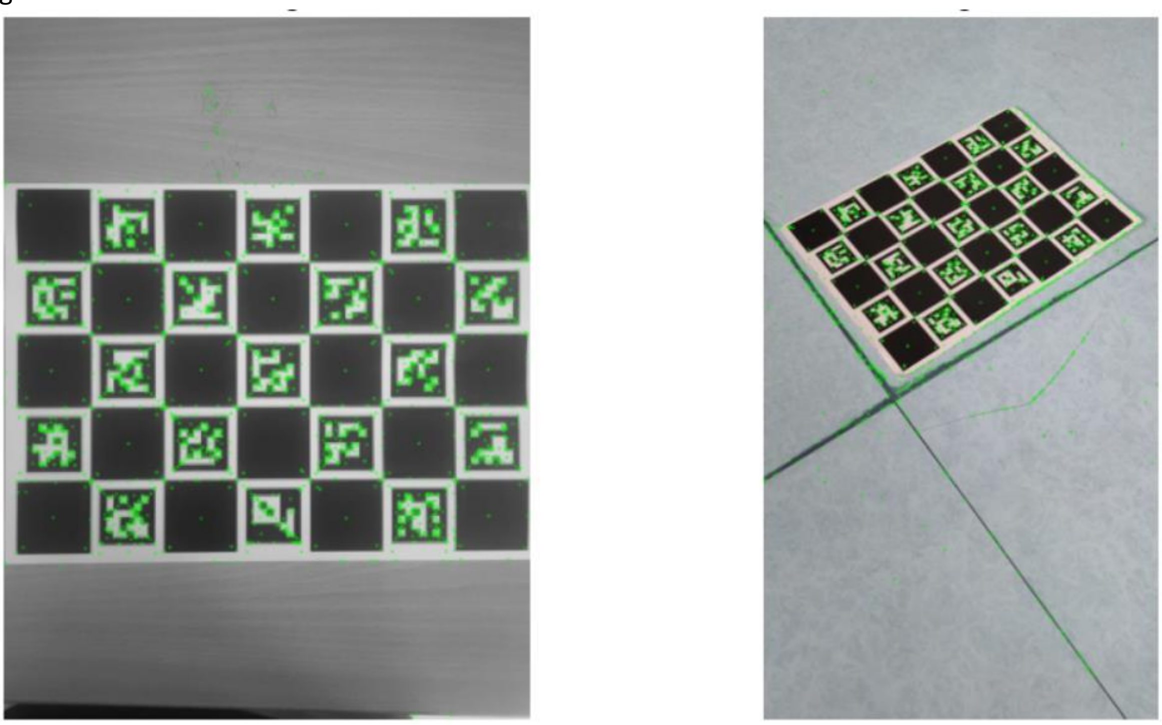


Figure 4: Interest points in both reference and target images

Step 4: Extract the descriptors for these interest points (SIFT feature descriptor in OpenCV is used here). Match these descriptions using the L2 Norm distance between the features.

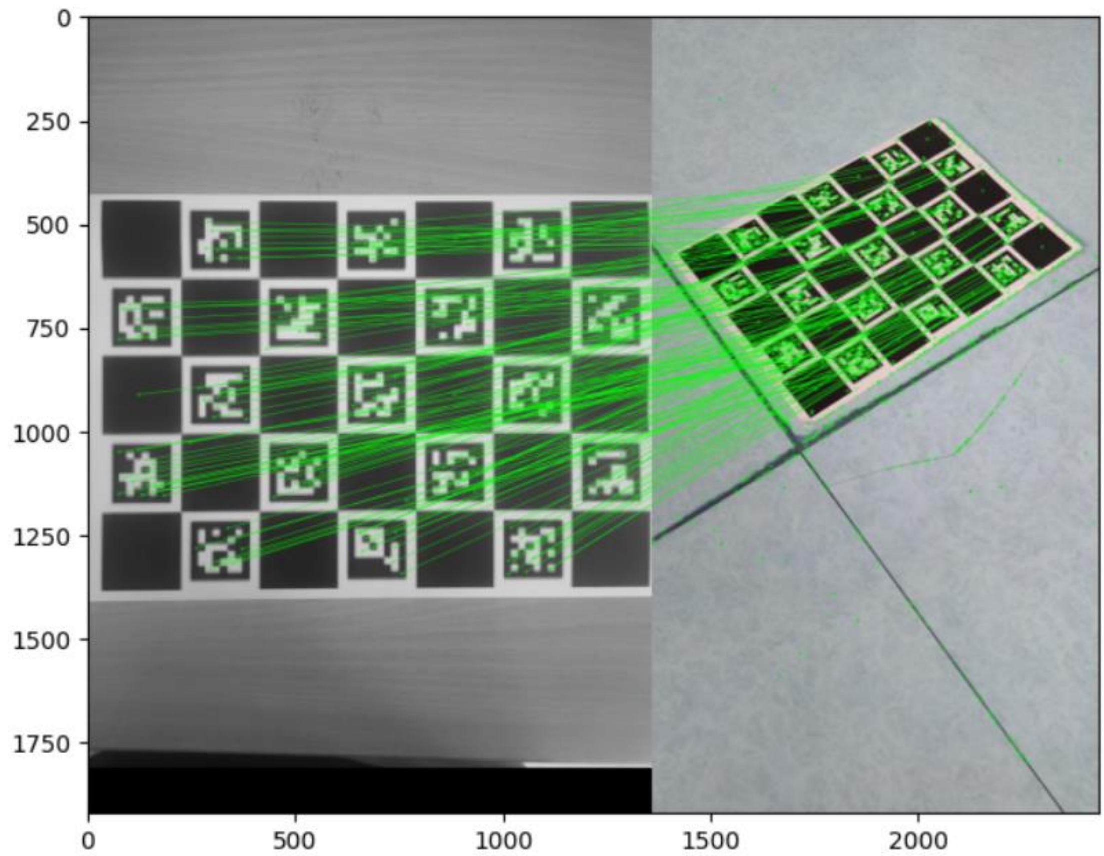


Figure 5: Matching the feature descriptors using the L2 norm distance between the features.

Step 5: Find the homography between the set of matched interest points using RANSAC and findHomography function in OpenCV.

Step 6: Find the projection matrix using the intrinsic parameters K and homography matrix H. (discussed in detail above in theory section.)

Step 7: Apply the projection matrix on the vertices of the 3D model and use cv2.fillConvexPoly to fit the 3D model in the image.

Step 8: Repeat Steps 1 to 7 for all the frames in the video to get the final output

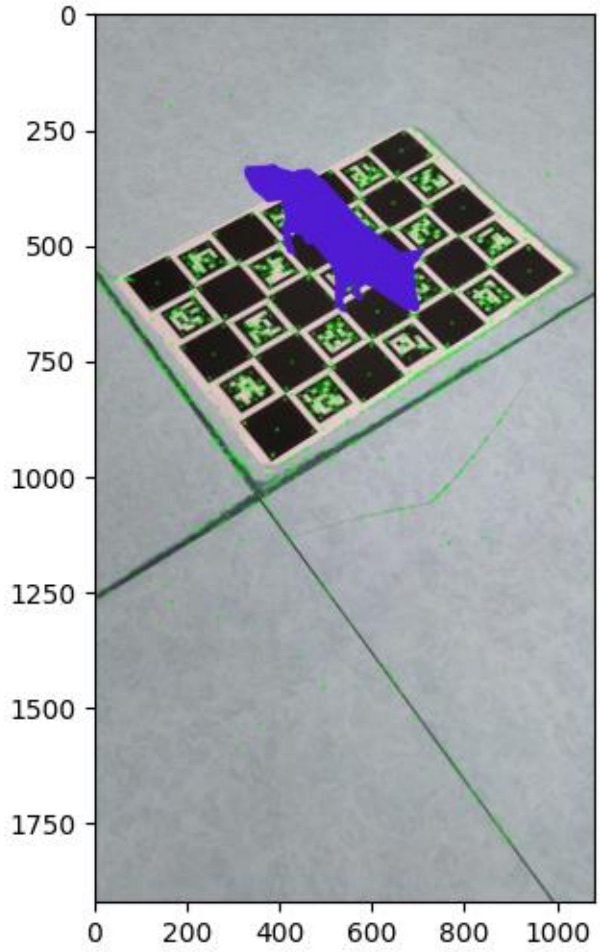


Figure 6: 3D model rendered on the target image

Results:

All the results for three 3D models wolf.obj, cat.obj and chair.obj can be found at

outputVideos

The github link for the code and output is also available at

For calibration : https://github.com/harikuttan7136/Camera-Calibration

For 3D rendering : https://github.com/harikuttan7136/Augmented-Reality

For the rat 3D model the result obtained for different frames is as below