COL - 780 Computer Vision - Assignment 3
Dharmisha Sharma(2022MCS2062)

I. **Calibrate the camera on your smartphone (intrinsic parameters) using the system using a checkerboard pattern as described in class.**

Algorithm for finding K (intrinsic parameters):

1. Click 6 images of a checkerboard from different angles from a smartphone.
2. Find 4 points in an image for a parallelogram on the checkerboard.
3. Using the cross product of these 4 points, find 4 lines (2 pairs of parallel lines).
4. For each pair of parallel lines, find a corresponding vanishing point using the cross product of the lines.
5. We'll get 2 vanishing points for each image i.e. 12 vanishing points for 6 images.
6. Let v_1, v_2 be the two vanishing points of an image.
7. Using the equation $v_1^T W v_2 = 0$, we can find W where W is a 3*3 matrix and $W = K^{-T} K^{-1}$
8. W is a symmetric matrix, therefore it has 6 unknowns, taking skew = 0, we have 5 unknowns.
9. Hence we can find W and get K.
10. Let, $v_1 = [x_1$
    y_1
    $z_1]$,
    $v_2 = [x_2$
    y_2
    $z_2]$,
    Therefore, $v_1^T = [x_1\ y_1\ 1]$
11. Using DLT, we'll get the following matrix
    $[\ x_{11}*x_{21}\quad z_{11}*x_{21}+x_{11}*z_{21}\quad y_{11}*y_{21}\quad z_{11}*b_{21}+b_{11}*z_{21}\quad z_{11}*z_{21}]$
    $[\ x_{12}*x_{22}\quad z_{12}*x_{22}+x_{12}*z_{22}\quad y_{12}*y_{22}\quad z_{12}*b_{22}+b_{12}*z_{22}\quad z_{12}*z_{22}]$
    $[\ x_{13}*x_{23}\quad z_{13}*x_{23}+x_{13}*z_{23}\quad y_{13}*y_{23}\quad z_{13}*b_{23}+b_{13}*z_{23}\quad z_{13}*z_{23}]$
    $[\ x_{14}*x_{24}\quad z_{14}*x_{24}+x_{14}*z_{24}\quad y_{14}*y_{24}\quad z_{14}*b_{24}+b_{14}*z_{24}\quad z_{14}*z_{24}]$
    $[\ x_{15}*x_{25}\quad z_{15}*x_{25}+x_{15}*z_{25}\quad y_{15}*y_{25}\quad z_{15}*b_{25}+b_{15}*z_{25}\quad z_{15}*z_{25}]$
    $[\ x_{16}*x_{26}\quad z_{16}*x_{26}+x_{16}*z_{26}\quad y_{16}*y_{26}\quad z_{16}*b_{26}+b_{16}*z_{26}\quad z_{16}*z_{26}]$
    Let this matrix be A.
12. Using SVD we can find W,
    $A = UDV^T$
    Where the last row of V^T depicts W and then after cholesky decomposition we can get K^{-T}
13. Taking inverse and transpose we can get K i.e. the intrinsic parameter matrix of a camera.
14. We can also verify the value of K calculated.
    K = [fx 0 ux]
    [0 fy uy]
    [0 0 1]
    The value of fx and fy should be almost equal and ux should be almost the half of max value of x coordinate of the image and uy should almost be equal to half of the max value of y coordinate of the image.

II. Placing the checkerboard on a table, insert some artificial objects on the table (such as a simple pyramid etc.) and generate some mixed/augmented-reality pictures. This need not be real-time but can be done on pictures taken from the smartphone.

Algorithm for finding P(Projection matrix)

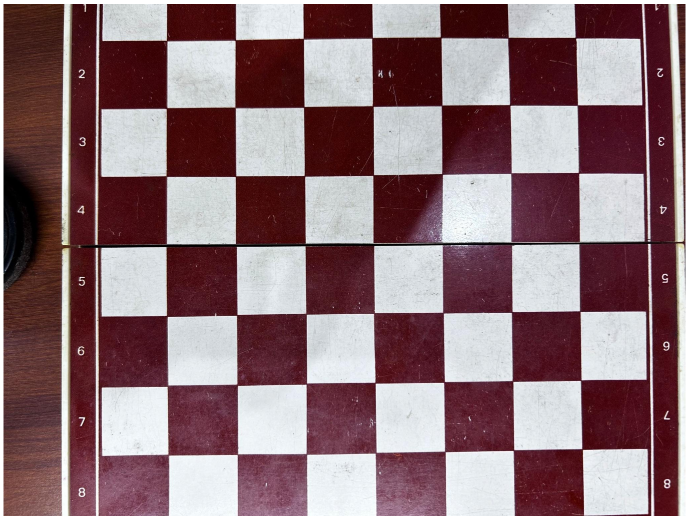
1. To find P i.e. the Projection matrix of a camera which is equal to K[R|t], we had already calculated K so we need to find R (Rotation matrix) and t(translation).
2. Click a picture of a checkerboard. Note the depth of the image with respect to the camera center because that will be the Z coordinate in the world coordinates of the image.
3. We can find P using x = PX where, x are the image coordinates and X are the real world coordinates.
4. x = PX   =>   x = K[R|t] X   => K^{-1} x = [R|t] X
5. Find 4 to 6 points(x) in the image and their corresponding world coordinates(X).
6. Let K^{-1} x = $[u \quad v \quad w]^T$
7. Using DLT we can get equations to calculate [R|t]

$$
\begin{bmatrix}
X_1 & Y_1 & Z_1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & X_1 & Y_1 & Z_1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 0 & 0 & 1 \\
X_2 & Y_2 & Z_2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & X_2 & Y_2 & Z_2 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & X_2 & Y_2 & Z_2 & 0 & 0 & 1 \\
X_3 & Y_3 & Z_3 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & X_3 & Y_3 & Z_3 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & X_3 & Y_3 & Z_3 & 0 & 0 & 1 \\
X_4 & Y_4 & Z_4 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & X_4 & Y_4 & Z_4 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & X_4 & Y_4 & Z_4 & 0 & 0 & 1
\end{bmatrix}
\times
\begin{bmatrix}
r_{11} \\ r_{12} \\ r_{13} \\ r_{21} \\ r_{22} \\ r_{23} \\ r_{31} \\ r_{32} \\ r_{33} \\ t_1 \\ t_2 \\ t_3
\end{bmatrix}
=
\begin{bmatrix}
u_1 \\ v_1 \\ w_1 \\ u_2 \\ v_2 \\ w_2 \\ u_3 \\ v_3 \\ w_3 \\ u_4 \\ v_4 \\ w_4
\end{bmatrix}
$$

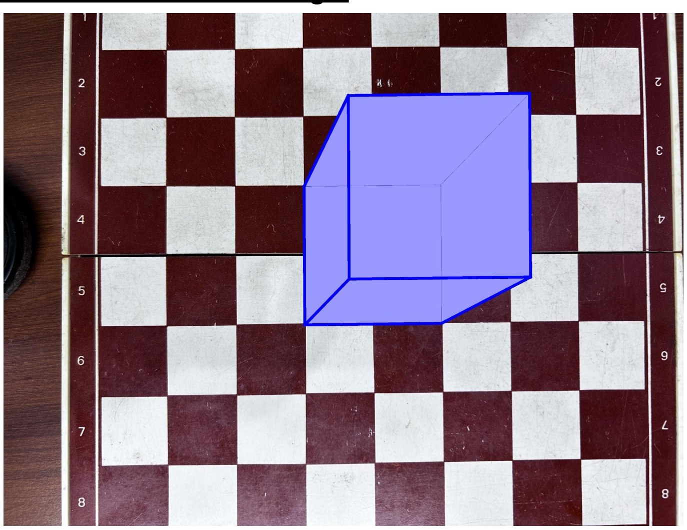
8. After getting [R|t] we can multiply it by K to get K [R|t] which is the projection matrix.
9. On getting the Projection matrix(P) we can insert an artificial object in the image by taking real world coordinates of that object and multiplying them by P to get the image coordinates for that image and displaying it on the image.
   Note: Since the image coordinates can be much much larger than the real world coordinates we can normalize the image coordinates while calculating the Projection matrix and later denormalize it when we want to display them on the image. In my case, I have used 1000 as the normalizing factor. While calculating P I have divided the x and y coordinates by 1000 and during Step 8 once I get the x and y coordinates I multiply them by 1000 to denormalize them.

Results:

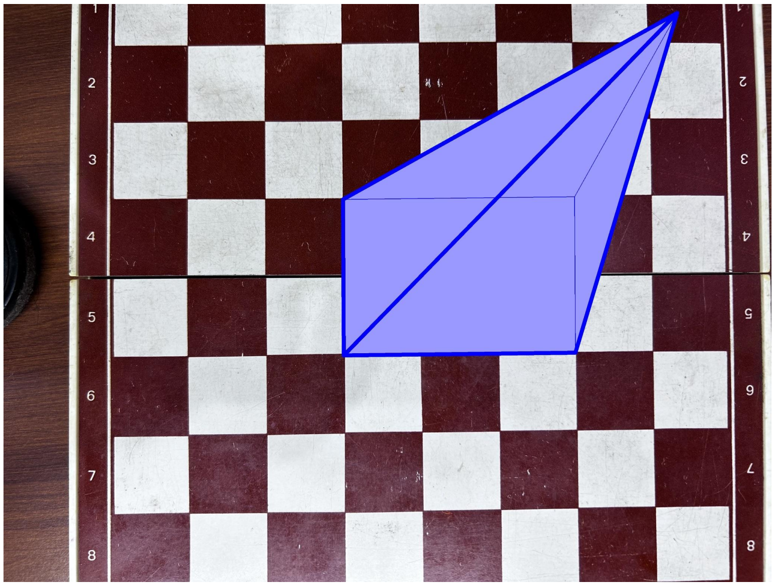
Original image:



Cube inserted in the image:



Pyramid inserted in the image:



Both cube and pyramid inserted in the image: