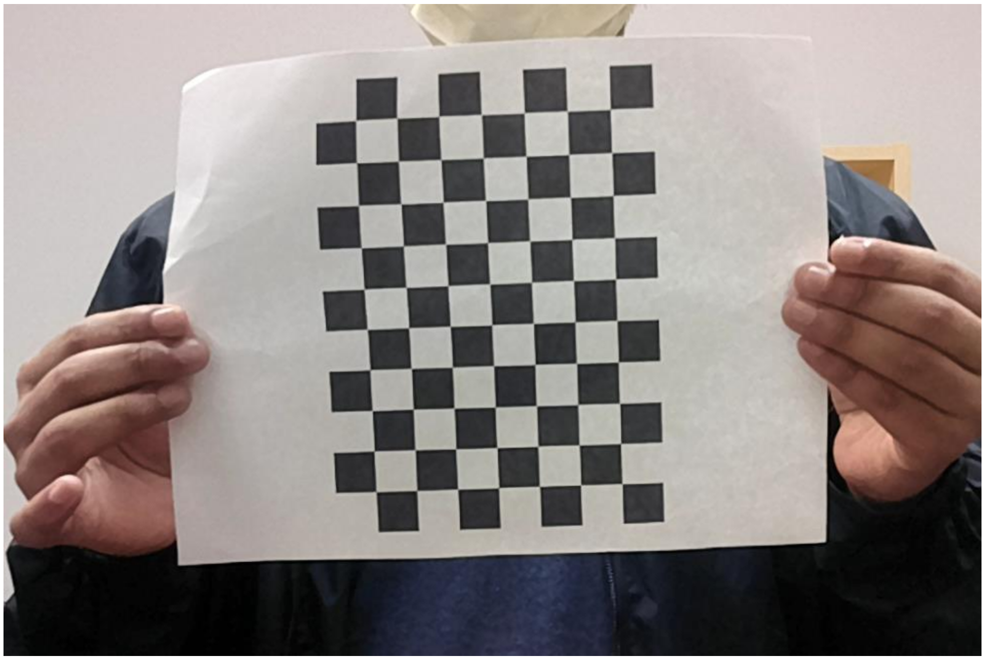
PART A: Theory

1.



An image of a checkerboard pattern is captured using the OAK-D Lite camera. The known focal length for the camera is 3.37 millimeter and the length from the camera to the actual object is 644 millimeters. The image resolution is 1080×720, which means that the image center lies at $(540, 360)$. However, we obtain u_0 and v_0 as 540 and 332 respectively. This shows that the image center does not coincide with our pre-set principal point. As estimated, it is offset by $(0, 28)$. We know that the camera matrix can be parameterized as:

$$P = \overbrace{K}^{\text{Intrinsic Matrix}} \times \overbrace{[R \mid \mathbf{t}]}^{\text{Extrinsic Matrix}}$$

$$= \overbrace{\underbrace{\begin{pmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{pmatrix}}_{\text{2D Translation}} \times \underbrace{\begin{pmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\text{2D Scaling}} \times \underbrace{\begin{pmatrix} 1 & s/f_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\text{2D Shear}}}^{\text{Intrinsic Matrix}} \times \overbrace{\underbrace{\left(I \mid \mathbf{t} \right)}_{\text{3D Translation}} \times \underbrace{\left(\begin{array}{c|c} R & 0 \\ \hline 0 & 1 \end{array} \right)}_{\text{3D Rotation}}}^{\text{Extrinsic Matrix}}$$

For the intrinsic matrix, we can calculate by using the formula below:

$$K = \begin{pmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$= \underbrace{\begin{pmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{pmatrix}}_{\text{2D Translation}} \times \underbrace{\begin{pmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\text{2D Scaling}} \times \underbrace{\begin{pmatrix} 1 & s/f_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\text{2D Shear}}$$

In the formula, f_x and f_y are the focal length, s is the skew factor, x_0 and y_0 are the offset values. Therefore, we can generate the intrinsic matrix as:

$$K = \begin{pmatrix} 3.37 & 0 & 0 \\ 0 & 3.37 & 28 \\ 0 & 0 & 1 \end{pmatrix}$$

For the extrinsic matrix, we need to compute the translation and rotation of the image compared to the expected position by using formula:

$$[R \mid t] = \left[\begin{array}{ccc|c} r_{1,1} & r_{1,2} & r_{1,3} & t_1 \\ r_{2,1} & r_{2,2} & r_{2,3} & t_2 \\ r_{3,1} & r_{3,2} & r_{3,3} & t_3 \end{array}\right]$$

As we known, the image has rotation through y-axis within 15 degrees (i.e., $\pi/12$). Therefore, we can generate the rotation matrix along y-axis as:

$$R = \begin{bmatrix} \cos(\pi/12) & 0 & -\sin(\pi/12) \\ 0 & 1 & 0 \\ \sin(\pi/12) & \cdots & \cos(\pi/12) \end{bmatrix} = \begin{bmatrix} 0.96592 & 0 & -0.25882 \\ 0 & 1 & 0 \\ 0.25882 & 0 & 0.96592 \end{bmatrix}$$

We can also generate the translation matrix by applying the offset values:

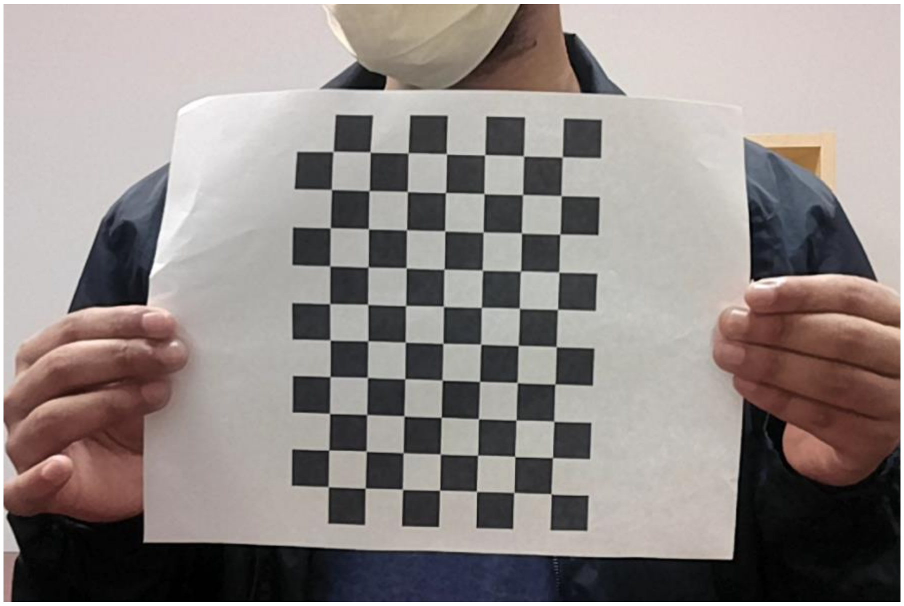
$$T = \begin{bmatrix} 0 \\ 28 \\ 0 \end{bmatrix}$$

In general, the extrinsic matrix can be generated as:

$$[R \mid t] = \begin{bmatrix} 0.96592 & 0 & -0.25882 & 0 \\ 0 & 1 & 0 & 28 \\ 0.25882 & 0 & 0.96592 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.

For finding the homography, we choose another image of the same plane but within a different angle of view:



Assume (x_1, y_1) is a point from the first image and $(\hat{x}_1, \hat{y}_1)$ is the corresponding same point in the second image. Then, we can say that these two points are related by an estimated homography H as:

$$\begin{bmatrix} \hat{x}_1 \\ \hat{y}_1 \\ 1 \end{bmatrix} = \frac{1}{z_a} \ H_{3\times 3} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

We can further parametrize the $H_{3\times 3}$ matrix as:

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$$

In such case, we can evaluate the overall homography matrix as:

$$\begin{bmatrix} \hat{x}_i z_a \\ \hat{y}_i z_a \\ z_a \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

By choosing a set of four points in the first image: [684, 530], [728, 529], [685, 574], [728, 572], we can find their corresponding points in the second image: [661, 591], [704, 592], [660, 633], [702, 634]. Therefore, we can obtain the homography matrix by implementing a homography calculation function from cv2 as:

```
In [14]: # Four corners in source image
         pts_src = np.array([[684, 530], [728, 529], [685, 574],[728, 572]])

         # Four corners in destination image
         pts_dst = np.array([[661, 591],[704, 592],[660, 633],[702, 634]])

         # Calculate Homography
         h, status = cv2.findHomography(pts_src, pts_dst)

In [15]: h

Out[15]: array([[ 4.30592128e-01, -2.07816394e-02,  2.00839446e+02],
                [-2.06908950e-01,  7.09575758e-01,  1.98508004e+02],
                [-4.04476587e-04,  1.77654352e-05,  1.00000000e+00]])
```

PART B: MATLAB Prototyping

```
 I = imread("./captured_images/1651007099257.jpg");
imshow(I);

% Get image coordinates
[x y] = ginput(2)
```

```
x = 2×1
   320
   892
y = 2×1
   430
   406
```

```
% Get focal length of OAK-D Lite from DepthAI
f = 1.6574233e+03;

% Distance between object and chessboard
z0 = 644;

% Calculate real world coordinates from image coordinates
x0 = z0 * (x(1) / f)
```

x0 = 124.3376

```
x1 = z0 * (x(2)/ f)
```

x1 = 346.5910

```


y0 = z0 * (y(1) / f)
```

y0 = 167.0786

```
y1 = z0 * (y(2) / f)
```

y1 = 157.7533

```
% Print out the distance of object
distance = sqrt((x1-x0)^2 + (y1 - y0)^2)
```

distance = 222.4490

Validate:
The actual object distance is 166 millimeters.

PART C: Application development

Link to the github repository:
https://github.com/annieee6446/CSC-8830-Computer-Vision-HW1