Homework 2

QUESTION 1 (15 MARKS)

1. (4 MARKS) 2. (3 MARKS) 3. (4 MARKS) 4.(4 MARKS)

Question. 1

1. (4 marks) Coordinate Transformation matrix

$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$R_y(\pi/2) = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix}$$

$$R_x(-\pi/2) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$

$$R_x(-\pi/2) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$T(-1,3,2) = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Overall Coordinate Transformation matrix.

$$T \times [R_x(-\pi/2)] \times [R_y(\pi/2)]$$

* we add an extra row and column to Rx and Ry to ensure same dimensionality during matrix multiplication

$$\Rightarrow \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 0 & 1 & 3 \\ 0 & -1 & 0 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\Rightarrow \boxed{\begin{bmatrix} 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 3 \\ 0 & -1 & 0 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}}$$

1.2 (3points) New coordinates of this vector

$X = (2,5,1)^T$ $\quad A = \begin{bmatrix} 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 3 \\ 0 & -1 & 0 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Let the new pt. be x'

$x' = Ax$

$$x' = \begin{bmatrix} 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 3 \\ 0 & -1 & 0 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 5 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ -3 \\ 1 \end{bmatrix}$$

$\boxed{\text{New point} = (0,1,-3)}$

1.3 (4 points)

$$R_y(\pi/2) = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix}$$

$$R_x(-\pi/2) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}$$

Final Rotated Axis = $R_x \circ R_y = R_c$

$$R_c = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{bmatrix}$$

- Characteristic Polynomial $= 1 - \lambda^3$
- Eigenvalues $\Rightarrow$ $\lambda_1 = 1$

$$\lambda_2 = \frac{1}{2}(-1 + i\sqrt{3}) \quad \lambda_3 = \frac{1}{2}(-1 - i\sqrt{3})$$

- Eigenvectors $\Rightarrow$ $V_1 = (-1, 1, 1)$

$$V_2 = \left(\frac{1}{2}(1 - i\sqrt{3}), \frac{1}{2}(-1 - i\sqrt{3}), 1 \right)$$

$$V_3 = \left(\frac{1}{2}(1 + i\sqrt{3}), \frac{1}{2}(-1 + i\sqrt{3}), 1 \right)$$

The eigenvector $V_1 = [-1, 1, 1]$ corresponds to the axis of rotation.

Normalization $\Rightarrow \frac{1}{\sqrt{3}}[-1, 1, 1]$

now, we have $\hat{V}_1$ (direction of the axis of roto)

angle of rotation $(\theta) = \arccos\left(\frac{\text{trace}(R_c) - 1}{2}\right)$

$\text{trace}(R_c) = 0$

$\cos\theta = -\frac{1}{2}$ $\boxed{\theta = 2\pi/3}$

$$\boxed{\begin{aligned}\text{Direction of axis of rotation} &= \left[\frac{-1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right] \\ \text{Angle of Rotation} &= 2\pi/3\end{aligned}}$$

1.4 (4 points)

1. We will find the combined matrix, we will get by sequentially applying the rotations.

$$R_x \times R_y = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix}$$

2. We have the final axis of rotation

$$\text{axis} = \frac{1}{\sqrt{3}}\left[-1, 1, 1\right]$$

$$\text{angle} = 2\pi/3$$

We will use the Rodrigues's formula to find the combined rotation matrix.

- $u_x = -1/\sqrt{3}$ $\quad u_y = 1/\sqrt{3}$ $\quad u_z = 1/\sqrt{3}$

- $$k = \begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix} \times \frac{1}{\sqrt{3}}$$

- $\sin(2\pi/3) = \sqrt{3}/2$ $\quad \cos(2\pi/3) = -1/2$

$$R = I + \sin(2\pi/3) \cdot k + (1 - \cos(2\pi/3))\, k^2$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & -1/2 & 1/2 \\ 1/2 & 0 & 1/2 \\ -1/2 & -1/2 & 0 \end{bmatrix} + \begin{bmatrix} -1 & -1/2 & -1/2 \\ -1/2 & -1 & 1/2 \\ -1/2 & 1/2 & -1 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{bmatrix}$$

Final matrix through sequential rotation and then multiplying the 2 rotation matrices.

$$\begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{bmatrix}$$

Final matrix obtained after applying Rodrigues's formula.

$$\begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{bmatrix}$$

Hence, shown we can achieve the same result.

QUESTION 2 (5 MARKS)

Q.2 (5 points) Rotation R, angle Θ about an axis u.

To show $\cos\Theta \cdot x + \sin\Theta\,(u \times (x)) + (1-\cos\Theta)(u^T x)\,u = Rx$

We know Rodrigues' formula

$$R = I + (\sin\Theta)\,k + (1-\cos\Theta)\,k^2$$

where

$$k = \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix}$$

Axis of rotation = $[u_x \;\; u_y \;\; u_z]$

Step 1 ⇒ We will show that
$$k^2 = -(u \cdot u)\,I + uu^T$$

$$k^2 = \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix} \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} -(u_z^2+u_y^2) & u_x u_y & u_x u_z \\ u_x u_y & -(u_z^2+u_x^2) & u_z u_y \\ u_z u_x & u_z u_y & -(u_y^2+u_x^2) \end{bmatrix}$$

$\Rightarrow (u \cdot u)\,I - uu^T$

$$[u_x^2+u_y^2+u_z^2] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} [u_x \;\; u_y \;\; u_z]$$

Date

$$\Rightarrow (u_x^2+u_y^2+u_z^2)I - \begin{bmatrix} u_x^2 & u_xy & u_xz \\ u_xy & u_y^2 & u_yz \\ u_xz & u_yz & u_z^2 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} (u_y^2+u_z^2) & -u_xy & -u_xz \\ -u_xy & (u_x^2+u_z^2) & -u_yz \\ -u_xz & -u_yz & (u_x^2+u_y^2) \end{bmatrix}$$

Hence Proved,

$$k^2 = uu^T - (u \circ u) I$$

Since u is a unit vector

$$u \circ u = 1$$

$$k^2 = uu^T - I$$

We will put this in the Rodrigue's eq^n

$$R = I + (\sin\theta)k + (1-\cos\theta)\underline{\underline{k^2}}$$

$$= I + (\sin\theta)k + (1-\cos\theta)(uu^T - I)$$

$$= I\cos\theta + (\sin\theta)k + (1-\cos\theta)(uu^T)$$

<u>step 2</u> $\Rightarrow$ Multiply by x on both LHS & RHS

$$Rx = Ix\cos\theta + (\sin\theta)kx + (1-\cos\theta)(uu^T)x$$

Date

Step3 ⇒ Show that

$$Kx = u \times (x)$$

LHS

$$\begin{bmatrix} 0 & -U_z & U_y \\ U_z & 0 & -U_x \\ -U_y & U_x & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$\begin{bmatrix} U_y.z - U_z y \\ U_z.x - U_x.x \\ U_x.x - U_y.x \end{bmatrix}$$

RHS

$$U = [U_x \quad U_y \quad U_z]$$

$$X = [x \quad y \quad z]$$

$$\Rightarrow \begin{bmatrix} U_y.z - U_z.y \\ U_z.x - U_x.x \\ U_x.x - U_y.x \end{bmatrix}$$

LHS = RHS

Hence, Shown

So, our equation becomes

$$R_x = Ix\cos\theta + (\sin\theta)(u \times (x)) + (1-\cos\theta)(uu^T)x$$

Step4 ⇒ Show that

$$(uu^T)x = (u^T x)u$$

⇒ First we will find $u^T x$

dot product of u^T and x

$$u^T x = (U_x \quad U_y \quad U_z)\begin{pmatrix} x \\ y \\ z \end{pmatrix} = U_x.x + U_y.y + U_z.z$$

Next, we will find

$$uu^T x = u(u^T x)$$

$$U(U_x \cdot x + U_y \cdot y + U_z \cdot z)$$

$$\begin{bmatrix} U_x \\ U_y \\ U_z \end{bmatrix} \begin{bmatrix} U_x \cdot x + U_y \cdot y + U_z \cdot z \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} U_x^2 \cdot x + U_x U_y \cdot y + U_x U_z \cdot z \\ U_x U_y \cdot x + U_y^2 \cdot y + U_z U_y \cdot z \\ U_x U_z \cdot x + U_y \cdot U_z \cdot y + U_z^2 \cdot z \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} (U^T x)\, U_x \\ (U^T x)\, U_y \\ (U^T x)\, U_z \end{bmatrix} = (U^T x)(u)$$

Hence, shown

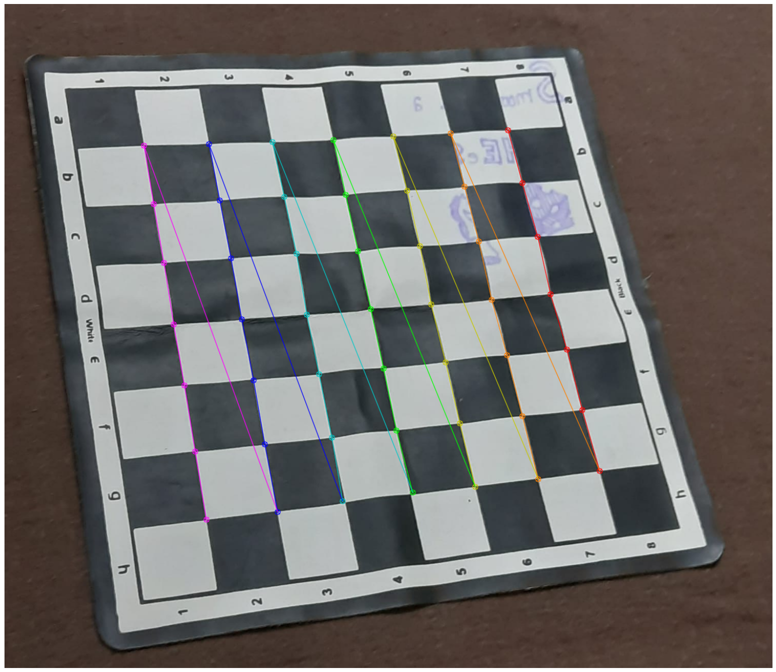
$$\Rightarrow Rx = Ix\cos\theta + (\sin\theta)(U \times (x)) + (1-\cos\theta)(u^T x U)$$

Also since $Ix = x$

$$\boxed{Rx = x\cos\theta + (\sin\theta)\, U \times (x) + (1-\cos\theta)(U^T x U)}$$

— Hence Proved

QUESTION 4 (40 MARKS)



1. (5 MARKS)

```
Estimated camera matrix (Intrinsic parameters):
[[1.26055760e+03 0.00000000e+00 6.66260556e+02]
 [0.00000000e+00 1.25537104e+03 7.42248654e+02]
 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]

Reprojection error:  0.6021381827654988
```

Focal lengths (pixels)

```
Focal Length (fx, fy): 1260.5575973041807 1255.3710410524163
```

Skew Parameter (pixels)

```
Skew Parameter (s): 0.0
```

Principal Point (pixels)

```
Principal Point (px, py): 666.2605556082682 742.2486537259402
```

Error Estimates

```
Reprojection error:  0.6021381827654988
```

```
Mean Squared Error (MSE): 0.48865539357326715
Maximum Reprojection Error (MRE): 1.4525326186233287
```

Maximum Reprojection Error (MRE) denotes the largest discrepancy observed across all image points. It shows the worst case error estimate in pixels.

2. (5 MARKS)

```
Image,Rotation Vector,Translation Vector
1,[-0.17492389 -0.07053249 -0.00442459],[-17.14062938 -16.26418
2,[-0.04770052 -0.00802384 -0.00130598],[-15.39795709 -13.77967
3,[-0.28604338 -0.12223497  0.05393227],[-14.88745384 -23.79354
4,[ 0.07659033 -0.09154895  0.02373859],[-13.99621401  -9.67194
5,[-0.22085241  0.37210697  1.50857676],[ 10.27766973 -14.96426
6,[-0.29601004  0.43140749  1.52994597],[ 10.29838802 -15.94440
7,[0.41448804 0.45397788 1.54478541],[  2.36413987 -12.25934907
8,[-0.60992914  0.05541     0.08121889],[ -7.13138328 -23.69141
9,[ 0.06323511 -0.04743043  1.5591875 ],[ 19.9597619  -18.04206
10,[-0.32252647 -0.50908201  1.03840346],[  8.03999123 -13.1185
11,[ 0.0554875  -0.1376595  -0.05562282],[  1.48308681 -19.4029
12,[-0.15616141  0.78606714  2.37256094],[ 26.01716634 -12.6424
13,[0.01211167 0.68905423 1.57463676],[  8.26368965 -25.244797
14,[-0.0661319  -0.04407313  0.01483368],[-17.63641398 -12.0137
15,[-0.06862798 -0.07964263 -0.00930254],[-19.46840805 -11.1147
16,[-0.20092709  0.67622111  1.43090742],[ 10.75853944 -25.6130
```

```
17,[0.15617088 0.33431476 1.59158738],[ 10.00898268 -20.3759567
18,[-0.07108181  0.13271414  1.56686564],[ 10.13995969 -13.4434
19,[-0.02073823 -0.12593326  0.00456435],[-17.45227942 -16.5990
20,[-0.26626008  0.40846138  1.5482122 ],[ 10.31738917 -13.3614
21,[-0.0466646   0.11250173  1.55722246],[ 10.80614761 -13.1887
22,[-0.06883555  0.13831892  1.54859325],[ 10.01550709 -13.6000
23,[0.01184844 0.21857336 1.57360547],[ 10.0265109  -13.3030706
24,[-0.17318992 -0.07066697 -0.02643947],[-16.614673   -11.4193
25,[-0.16779952 -0.21476329 -0.06294618],[-13.78547401 -11.0792
26,[0.04269543 0.12391427 1.56589941],[ 10.65688454 -15.1058450
27,[0.11581965 0.21428565 1.5476829 ],[  8.33067982 -16.074662
28,[0.22109023 0.29394234 1.53670442],[  6.1660072  -15.614204
29,[-0.32307275 -0.16519501 -0.06568346],[-18.41200158 -20.2359
30,[-0.21404609  0.39974099  1.52812255],[ 10.18690118 -19.5320
31,[-0.22268218  0.40041774  1.53490001],[ 10.19842586 -19.4728
32,[-0.35346588  0.56847914  1.59291788],[ 20.62982073 -22.3492
33,[-0.33984196  0.48865354  1.58259961],[ 15.48672013 -28.6441
34,[-0.14828613  0.25258994  1.5259116 ],[ 17.62296728 -18.3836
35,[-0.30851816  0.45475649  1.56723512],[ 19.34527552 -22.4925
36,[-0.27996875  0.28996061  1.5158099 ],[  9.29848067 -12.3913
37,[-0.3847194   0.07851839  1.40290974],[ 12.84384858 -20.1827
```

Example

First image has,

Rotation Vector = [-0.17492389 -0.07053249 -0.00442459]

- Negative Pitch = Pitch refers to the rotation of the camera up or down relative to a horizontal plane. Camera is tilted upwards
- Small Negative Yaw = Yaw refers to the rotation of the camera left or right around its vertical axis. Camera is rotated to the left.
- Very Small Negative Roll = Roll refers to the rotation of the camera clockwise or counterclockwise around its optical axis. Almost no clockwise or anti clockwise rotation.

Translation Vector = [-17.14062938 -16.26418077 54.79242835]

- Camera is on the left of the image's center by 17 cm.
- Camera is below the image's center by 16 cm.
- Camera is above the image's center by 54.8 cm.

3. (5 MARKS)

```
Radial Distortion Coefficients:
k1: [ 0.0927011  -0.51595751 -0.01754788 -0.00374097  0.47325152]
```

Undistorted vs Original



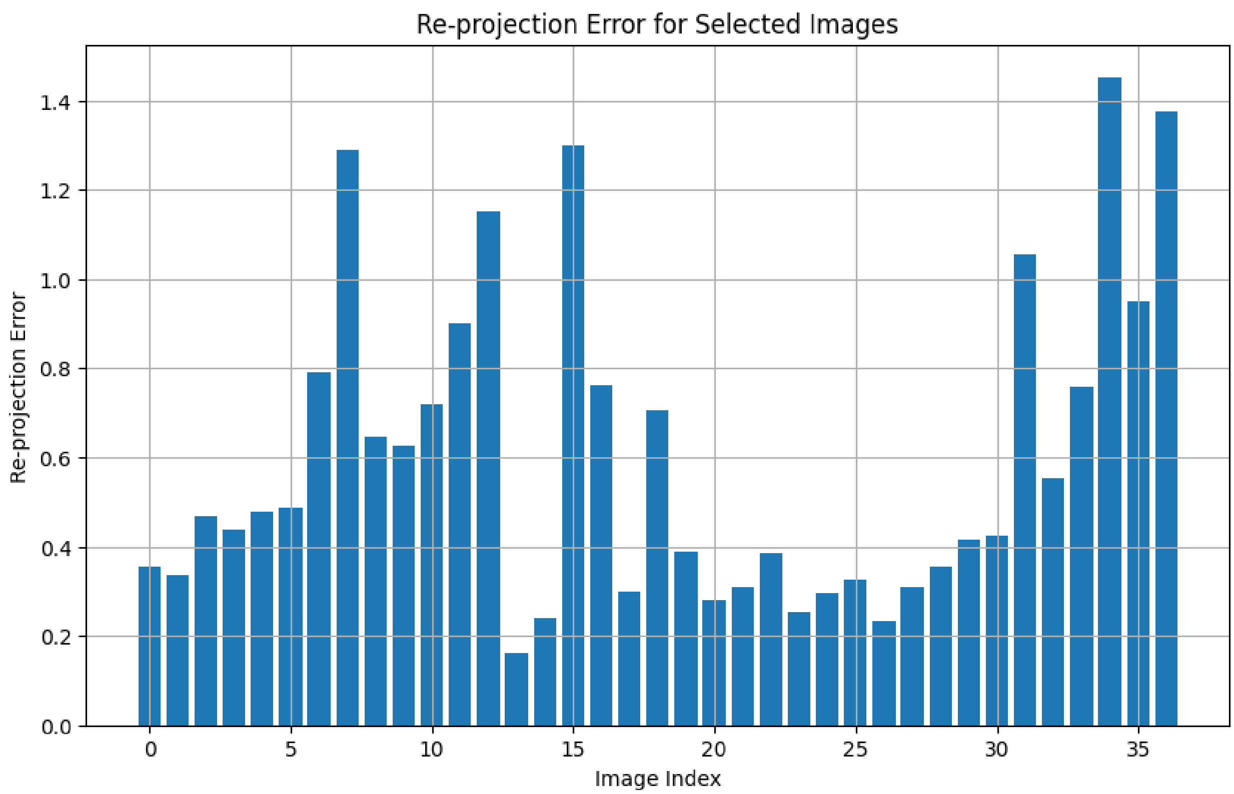




Observations —

- The lines at the edges are more straight.
- In the original image, the lines at the edges appear to be curved inside, but in the undistorted image, that is resolved.

4. (5 MARKS)



Mean Re-projection Error: 0.6021381827654988
Standard Deviation of Re-projection Error: 0.35508450040677664

5. (10 MARKS)

Image with Detected and Re-projected Corners

Image with Detected and Re-projected Corners

Image with Detected and Re-projected Corners

Image with Detected and Re-projected Corners

Image with Detected and Re-projected Corners

Image with Detected and Re-projected Corners

Image with Detected and Re-projected Corners

Image with Detected and Re-projected Corners

Image with Detected and Re-projected Corners

Image with Detected and Re-projected Corners

Image with Detected and Re-projected Corners

Reprojection Error Calculation

The reprojection error is computed by comparing the detected image points (corners) with the re-projected image points.

Here's a breakdown of how it's computed in the provided code:

1. Calculate reprojection error by comparing detected image points with re-projected ones using camera calibration parameters.
2. Iterate over each image in the dataset.
3. Reproject object points onto the image plane using calibration parameters.
4. Compute the normalized L2 (Euclidean) norm between detected and reprojected points.
5. Append normalized error for each image to the errors list.
6. Return the list of reprojection errors for all images.
7. Plot corners using the plot corner function.
8. Take image as input along with object points, camera matrix, distortion coefficients, rotation vectors, and translation vectors.
9. Read and convert the image to grayscale.
10. Detect corners in the image.
11. Refine corner positions if detected.
12. Reproject object points onto the image plane.

13. Draw detected and reprojected corners on the original image.
14. Display the image.

The reprojection error provides a measure of how accurately the camera calibration parameters (intrinsic and extrinsic) can reproduce the detected image points from the known object points. Lower reprojection errors indicate better calibration accuracy.

6. (10 MARKS)

```
Image 1 - Checkerboard Plane Normal: [-0.06972899  0.17404364  
Image 2 - Checkerboard Plane Normal: [-0.00798957  0.04768714  
Image 3 - Checkerboard Plane Normal: [-0.12786415  0.27804639  
Image 4 - Checkerboard Plane Normal: [-0.09041518 -0.07748669  
Image 5 - Checkerboard Plane Normal: [0.10201886 0.36831681 0.9
Image 6 - Checkerboard Plane Normal: [0.08531311 0.44696994 0.8
Image 7 - Checkerboard Plane Normal: [0.52477195 0.02871147 0.8
Image 8 - Checkerboard Plane Normal: [0.02796082 0.57404274 0.8
Image 9 - Checkerboard Plane Normal: [ 0.00968434 -0.07055307  
Image 10 - Checkerboard Plane Normal: [-0.54357906  0.01638502
Image 11 - Checkerboard Plane Normal: [-0.13862378 -0.05143496
Image 12 - Checkerboard Plane Normal: [0.08022205 0.57349362 0.
Image 13 - Checkerboard Plane Normal: [0.40390462 0.41445192 0.
Image 14 - Checkerboard Plane Normal: [-0.04451536  0.06573318
Image 15 - Checkerboard Plane Normal: [-0.07917594  0.06887074
Image 16 - Checkerboard Plane Normal: [0.30800534 0.51542044 0.
Image 17 - Checkerboard Plane Normal: [0.30319856 0.11648996 0.
Image 18 - Checkerboard Plane Normal: [0.03921334 0.1293587  0.
Image 19 - Checkerboard Plane Normal: [-0.12563847  0.02039489
Image 20 - Checkerboard Plane Normal: [0.086646   0.41639981 0.
Image 21 - Checkerboard Plane Normal: [0.04250064 0.10104684 0.
Image 22 - Checkerboard Plane Normal: [0.04549501 0.13138126 0.
Image 23 - Checkerboard Plane Normal: [0.1450721  0.13125382 0.
Image 24 - Checkerboard Plane Normal: [-0.06796468  0.17309305
```

```
Image 25 - Checkerboard Plane Normal: [-0.20672681  0.17233521
Image 26 - Checkerboard Plane Normal: [0.10594896 0.05145718 0.9
Image 27 - Checkerboard Plane Normal: [0.20949979 0.06063657 0.9
Image 28 - Checkerboard Plane Normal: [0.32336317 0.04268656 0.9
Image 29 - Checkerboard Plane Normal: [-0.15098594  0.32116457
Image 30 - Checkerboard Plane Normal: [0.11899427 0.38024184 0.9
Image 31 - Checkerboard Plane Normal: [0.11251355 0.38591407 0.9
Image 32 - Checkerboard Plane Normal: [0.10686657 0.55280763 0.8
Image 33 - Checkerboard Plane Normal: [0.07627014 0.5024403  0.8
Image 34 - Checkerboard Plane Normal: [0.0704129  0.25235564 0.9
Image 35 - Checkerboard Plane Normal: [0.08165558 0.46663206 0.8
Image 36 - Checkerboard Plane Normal: [0.01276325 0.3567414  0.9
Image 37 - Checkerboard Plane Normal: [-0.1718392   0.30836765
```

QUESTION 5 (40 MARKS)

1. (10 MARKS)

```
Plane 1: Normal = [ 0.21290252  0.97660804 -0.03015424],
Offset = 0.00019648578017950058

Plane 2: Normal = [-0.27013198  0.9362082  -0.22481766],
Offset = 0.000760495662689209

Plane 3: Normal = [ 0.09349882 -0.06432354  0.99353933],
Offset = 0.00025666505098342896

Plane 4: Normal = [0.11655805 0.993106   0.01243492],
Offset = 0.00023660948500037193

Plane 5: Normal = [ 0.18587807 -0.90856564 -0.37410933],
Offset = 0.0005525946617126465
```

```
Plane 6: Normal = [0.4484332  0.48426893 0.7512598 ],
Offset = -0.0026053190231323242

Plane 7: Normal = [ 0.28584704  0.9416206  -0.17788222],
Offset = -0.0017397254705429077

Plane 8: Normal = [0.17063239 0.22785623 0.9586272 ],
Offset = 0.0016198158264160156

Plane 9: Normal = [ 0.07732306  0.34425956 -0.93568504],
Offset = 0.002199530601501465

Plane 10: Normal = [ 0.24815519  0.9415081  -0.22799441],
Offset = -0.0011558085680007935

Plane 11: Normal = [-0.23084207 -0.9729858   0.00324548],
Offset = 0.00011933152563869953

Plane 12: Normal = [-0.13441342 -0.04413806 -0.98994184],
Offset = 0.00037047266960144043

Plane 13: Normal = [-0.22794697 -0.971483   -0.06527653],
Offset = 0.00040764734148979187

Plane 14: Normal = [-0.20867775 -0.9748128  -0.07869963],
Offset = -0.0010669194161891937

Plane 15: Normal = [ 0.13746653  0.88259166 -0.4495942 ],
Offset = 0.004356712102890015

Plane 16: Normal = [-0.4700125  -0.8813593   0.04789697],
Offset = -0.0022859349846839905

Plane 17: Normal = [ 0.00189043 -0.2047629   0.9788098 ],
Offset = -0.003598337061703205
```

```
Plane 18: Normal = [ 0.02435743 -0.05382288  0.9982534 ],
Offset = -0.0007001757621765137

Plane 19: Normal = [ 0.41474172  0.79058635 -0.45051366],
Offset = -0.00014963746070861816

Plane 20: Normal = [-0.16768335  0.98544407  0.02796934],
Offset = 0.0003482867032289505

Plane 21: Normal = [ 0.59689486  0.78773075 -0.15230486],
Offset = 0.0028872936964035034

Plane 22: Normal = [0.27275833 0.93204284 0.2385351 ],
Offset = -0.0007227510213851929

Plane 23: Normal = [-0.034497    0.9982623   0.04777433],
Offset = -0.00044509628787636757

Plane 24: Normal = [ 0.09761643  0.87309134 -0.4776846 ],
Offset = 0.006773233413696289

Plane 25: Normal = [ 0.10957734 -0.00251862  0.9939751 ],
Offset = 0.0002389056608080864
```

2. (10 MARKS)

Q.5.2 We have n_i^C and n_i^L
derive ${}^{C}T_L = [\,{}^{C}R_L \mid {}^{C}t_L\,]$
Transformation from LIDAR to camera frame.

With respect to the LIDAR frame, the camera has undergone a translation and a rotation

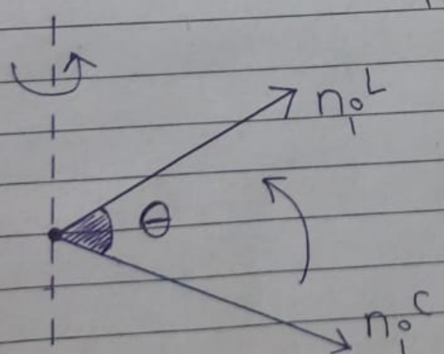
1. Angle b/w the planes = Angle b/w plane normals

$$n_i^C \circ n_i^L / |n_i^C||n_i^L| = \cos\theta$$

$$\boxed{\theta = \cos^{-1}(n_i^C \circ n_i^L)}$$

2. Axis of rotation

it will be ⊥ to both n_i^C and n_i^L



$$\boxed{\text{axis} = n_i^C \times n_i^L}$$

$$\hat{n} = n_i^C \times n_i^L$$

Using axis normal $(\hat{n})$, we can calculate k such that

$$k.x = \hat{n} \times (x) \quad \forall\, x \text{ vectors in } \mathbb{R}^3$$

3. Rotation matrix using Rodrigus' form.

$$R = I + (\sin\theta)k + (1-\cos\theta)k^2$$

$$\boxed{{}^{C}R_{L} = I + (\sin\theta)k + (1-\cos\theta)k^2}$$

The idea used in the code to calculate the translation matrix (CtL) is to find the centroid of both the camera frame and the LIDAR frame using the provided camera plane normals and LIDAR plane normals, respectively. Then, we compute the translation vector by subtracting the centroid of the LIDAR frame from the centroid of the camera frame. Finally, we construct the translation matrix using the translation vector.

Camera Plane = The camera is treated as a point located at the origin of the camera plane, facing towards the normal vector of the plane.

LIDAR Plane = LIDAR sensor as an object with a specific orientation and position relative to its surroundings. The direction in which the LIDAR sensor is facing can be represented by its normal vector, which is perpendicular to the plane of the sensor.

3. (5 MARKS)

4. (5 MARKS)

5. (10 MARKS)