CSE344: Computer Vision

Assignment-2

Divyajeet Singh (2021529)

March 20, 2024

Question 1.

1. The given transformation is a composition of the following three transformations (in order):

   (a) Rotation by $\frac{\pi}{2}$ about the Y-axis

   (b) Rotation by $\frac{-\pi}{2}$ about the X-axis

   (c) Translation by $t = \begin{bmatrix} -1 & 3 & 2 \end{bmatrix}^\top$

   The coordinate transformation matrices (using 3-dimensional homogeneous coordinates), for the three transformations are

$$R_y\left(\frac{\pi}{2}\right) = \begin{bmatrix} \cos\left(\frac{\pi}{2}\right) & 0 & \sin\left(\frac{\pi}{2}\right) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\left(\frac{\pi}{2}\right) & 0 & \cos\left(\frac{\pi}{2}\right) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_x\left(\frac{-\pi}{2}\right) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\left(\frac{-\pi}{2}\right) & -\sin\left(\frac{-\pi}{2}\right) & 0 \\ 0 & \sin\left(\frac{-\pi}{2}\right) & \cos\left(\frac{-\pi}{2}\right) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

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

   The coordinate transformation matrix for the full transformation is then given by

$$\begin{aligned} T &= T(t) \circ R_x\left(\frac{-\pi}{2}\right) \circ R_y\left(\frac{\pi}{2}\right) \\ &= T(t) \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 3 \\ 0 & -1 & 0 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

   since rotations and translations are linear transformations in homogeneous coordinate systems, and hence composition of transformations is equivalent to multiplication of the transformation matrices.

2. Now, we find the new coordinates of a given vector, $v = \begin{bmatrix} 2 & 5 & 1 \end{bmatrix}^\top$, after the transformation. The new coordinates, say v', are given by simply applying the transformation matrix to v in homogeneous coordinates, which gives

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

So, the new coordinates of the given vector are $v' = \begin{bmatrix}0 & 1 & -3\end{bmatrix}^\top$. We also find the point that the origin of the initial frame of reference gets mapped to. For this, we simply apply the transformation matrix T to the origin $\mathbf{0}$ in homogeneous coordinates, which gives

$$T\mathbf{0} = \begin{bmatrix} 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 3 \\ 0 & -1 & 0 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 3 \\ 2 \\ 1 \end{bmatrix}$$

So, the origin gets mapped to the point $\begin{bmatrix}-1 & 3 & 2\end{bmatrix}^\top$. In fact, the origin is invariant to pure rotations about the coordinate axes, so intuitively, the origin maps to the translation vector t.

3. We now have the combined rotation matrix, $\mathbf{R}$ (written without homogeneous coordinates), as

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

Using Rodrigues formula, we can obtain the direction of the axis of this combined rotation in the original frame of reference and the angle of rotation about this axis. According to the Rodrigues formula, the angle of rotation is given by

$$\theta = \cos^{-1}\left(\frac{\textsc{Trace}(\mathbf{R}) - 1}{2}\right) = \cos^{-1}\left(\frac{-1}{2}\right) = \frac{2\pi}{3}$$

The axis of rotation, $\hat{\mathbf{n}}$, is given by

$$\hat{\mathbf{n}} = \frac{1}{2\sin\theta} \begin{bmatrix} \mathbf{R}_{32} - \mathbf{R}_{23} \\ \mathbf{R}_{13} - \mathbf{R}_{31} \\ \mathbf{R}_{21} - \mathbf{R}_{12} \end{bmatrix} = \frac{1}{2\sin\frac{2\pi}{3}} \begin{bmatrix} -1 - 0 \\ 1 - 0 \\ -1 - 0 \end{bmatrix} = \frac{1}{\sqrt{3}} \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix}$$

Therefore, the combined rotation is a rotation of $\frac{2\pi}{3}$ about the axis $\frac{1}{\sqrt{3}}\begin{bmatrix}-1 & 1 & -1\end{bmatrix}^\top$, i.e.

$$\mathbf{R} \equiv R\left(\frac{1}{\sqrt{3}}\begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix}, \frac{2\pi}{3}\right)$$

4. We now use the above axis $\hat{\mathbf{n}}$ and angle θ to calculate the the rotation matrix (say) $\mathbf{R}'$ for the rotation, and show that it is the same as matrix $\mathbf{R}$ that we obtained through sequentially applying the two given rotations. Using the Rodrigues formula, the rotation matrix $\mathbf{R}'$ is given by

$$\mathbf{R}' = \mathbf{I} + \sin\theta\mathbf{N} + (1 - \cos\theta)\mathbf{N}^2 \quad \text{where} \quad \mathbf{N} = \begin{bmatrix} 0 & -n_3 & n_2 \\ n_3 & 0 & -n_1 \\ -n_2 & n_1 & 0 \end{bmatrix} \quad (n_i \text{ represent the components of } \hat{\mathbf{n}})$$

We first find $\mathbf{N}$ and then use it to find $\mathbf{R}'$. We have

$$\mathbf{N} = \frac{1}{\sqrt{3}} \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix} \implies \mathbf{N}^2 = \frac{1}{3} \begin{bmatrix} -2 & -1 & 1 \\ -1 & -2 & -1 \\ 1 & -1 & -2 \end{bmatrix}$$

Using these, we find

$$\begin{aligned}
\mathbf{R}' &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \frac{1}{\sqrt{3}}\sin\frac{2\pi}{3} \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix} + \frac{1}{3}\left(1 - \cos\frac{2\pi}{3}\right) \begin{bmatrix} -2 & -1 & 1 \\ -1 & -2 & -1 \\ 1 & -1 & -2 \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} -2 & -1 & 1 \\ -1 & -2 & -1 \\ 1 & -1 & -2 \end{bmatrix} \\
&= \begin{bmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} = \mathbf{R}
\end{aligned}$$

Hence, we have shown that the rotation matrix $\mathbf{R}'$ obtained using the axis $\hat{\mathbf{n}}$ and angle θ is the same as the matrix $\mathbf{R}$ that we obtained through sequentially applying the two given rotations. This also proves the correctness of the axis and angle obtained using the Rodrigues formula.

Question 2.

By Rodrigues formula, we know that the rotated vector for the given rotation is given by

$$\begin{aligned}
\mathbf{Rx} &= \mathbf{x} + (\hat{\mathbf{u}} \times \mathbf{x}) \sin\theta + (1 - \cos\theta)\hat{\mathbf{u}} \times (\hat{\mathbf{u}} \times \mathbf{x}) \\
&= \mathbf{x} + (\hat{\mathbf{u}} \times \mathbf{x}) \sin\theta + (1 - \cos\theta) \left[(\hat{\mathbf{u}}^\top \mathbf{x})\hat{\mathbf{u}} - (\hat{\mathbf{u}}^\top \hat{\mathbf{u}})\mathbf{x}\right] \\
&= \mathbf{x} + (\hat{\mathbf{u}} \times \mathbf{x}) \sin\theta + (1 - \cos\theta)(\hat{\mathbf{u}}^\top \mathbf{x})\hat{\mathbf{u}} - (1 - \cos\theta)\mathbf{x} \\
&= \mathbf{x} - \mathbf{x} + \mathbf{x}\cos\theta + (\hat{\mathbf{u}} \times \mathbf{x}) \sin\theta + (1 - \cos\theta)(\hat{\mathbf{u}}^\top \mathbf{x})\hat{\mathbf{u}} \\
&= \mathbf{x}\cos\theta + (\hat{\mathbf{u}} \times \mathbf{x}) \sin\theta + (\hat{\mathbf{u}}^\top \mathbf{x})(1 - \cos\theta)\hat{\mathbf{u}} \qquad \square
\end{aligned}$$

which proves the result, using the vector triple product identity that states for any vectors $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{c}$,

$$\mathbf{a} \times (\mathbf{b} \times \mathbf{c}) = (\mathbf{a}^\top \mathbf{c})\mathbf{b} - (\mathbf{a}^\top \mathbf{b})\mathbf{c}$$

and that $\hat{\mathbf{u}}^\top \hat{\mathbf{u}} = 1$ since $\hat{\mathbf{u}}$ is a unit vector.

Question 3.

We are given image formation equations of two cameras, $\mathbf{C}_1$ and $\mathbf{C}_2$, for the same 3D point $\mathbf{X}$ in the world in homogeneous coordinates, as follows

$$\begin{aligned}
\mathbf{x}_1 &= \mathbf{K}_1 [\mathbf{R}_1 \mid \mathbf{t}_1] \mathbf{X} \\
\mathbf{x}_2 &= \mathbf{K}_2 [\mathbf{R}_2 \mid \mathbf{t}_2] \mathbf{X}
\end{aligned}$$

where we assume that the extrinsic matrices $[\mathbf{R}_i \mid \mathbf{t}_i]_{3\times 4}$, include the perspective projection matrix $\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}$. We need to show that the image points are related by

$$\mathbf{x}_1 = \mathbf{H}\mathbf{x}_2$$

for some invertible $\mathbf{H}_{3\times 3}$ and find it in terms of $\mathbf{K}_1$, $\mathbf{K}_2$, and $\mathbf{R}$.
Given that the orientation of $\mathbf{C}_2$ is obtained by applying a 3D rotation $\mathbf{R}$ on $\mathbf{C}_1$, which means

$$[\mathbf{R}_2 \mid \mathbf{t}_2] = \mathbf{R} [\mathbf{R}_1 \mid \mathbf{t}_1]$$

Then, we have

$$\begin{aligned}
\mathbf{x}_1 &= \mathbf{H}\mathbf{x}_2 \\
\mathbf{x}_1 &= \mathbf{H}\mathbf{K}_2 [\mathbf{R}_2 \mid \mathbf{t}_2] \mathbf{X} \\
\mathbf{K}_1 [\mathbf{R}_1 \mid \mathbf{t}_1] \mathbf{X} &= \mathbf{H}\mathbf{K}_2\mathbf{R} [\mathbf{R}_1 \mid \mathbf{t}_1] \mathbf{X} \\
\implies \mathbf{K}_1 [\mathbf{R}_1 \mid \mathbf{t}_1] &= \mathbf{H}\mathbf{K}_2\mathbf{R} [\mathbf{R}_1 \mid \mathbf{t}_1] \quad \text{since } \mathbf{X} \text{ is arbitrary}
\end{aligned}$$

At this point, it is important to consider the shapes of the matrices involved. We have

$$\underbrace{\mathbf{K}_1}_{3\times 3} \underbrace{[\mathbf{R}_1 \mid \mathbf{t}_1]}_{3\times 4} = \underbrace{\mathbf{H}}_{3\times 3} \underbrace{\mathbf{K}_2}_{3\times 3} \underbrace{\mathbf{R}}_{3\times 3} \underbrace{[\mathbf{R}_1 \mid \mathbf{t}_1]}_{3\times 4}$$

Note that $[\mathbf{R}_1 \mid \mathbf{t}_1]$ is not invertible. However, we do not need to invert it. Since the operands to the right remain the same, it is sufficient to find $\mathbf{H}$ such that the

$$\begin{aligned}
\mathbf{K}_1 &= \mathbf{H}\mathbf{K}_2\mathbf{R} \\
\implies \mathbf{K}_1\mathbf{R}^\top &= \mathbf{H}\mathbf{K}_2 \\
\implies \mathbf{K}_1\mathbf{R}^\top\mathbf{K}_2^{-1} &= \mathbf{H}
\end{aligned}$$

Therefore, the matrix $\mathbf{H} = \mathbf{K}_1\mathbf{R}^\top\mathbf{K}_2^{-1}$ gives the required relation between the image points. Finally, we show two properties of $\mathbf{H}$

1. $\mathbf{H}$ is of size 3×3. This is easy to see as

$$\underbrace{\mathbf{H}}_{3\times 3} = \underbrace{\mathbf{K}_1}_{3\times 3} \underbrace{\mathbf{R}^\top}_{3\times 3} \underbrace{\mathbf{K}_2^{-1}}_{3\times 3}$$

2. $\mathbf{H}$ is invertible. This is also easy to see as

$$\mathbf{H}^{-1} = \left(\mathbf{K}_1 \mathbf{R}^\top \mathbf{K}_2^{-1}\right)^{-1} = \mathbf{K}_2 \mathbf{R} \mathbf{K}_1^{-1}$$

   where we crucially use the fact that $\mathbf{R}$ is an orthogonal matrix, and $\mathbf{K}_1$ and $\mathbf{K}_2$ are invertible since they are upper triangular, with non-zero entries ((possibly scaled) focal lengths) on their principal diagonals.

We can verify the correctness of the result by substituting the obtained $\mathbf{H}$ into the required relation. We have

$$\begin{aligned}
\mathbf{x}_1 &= \mathbf{H}\mathbf{x}_2 \\
&= \mathbf{K}_1 \mathbf{R}^\top \mathbf{K}_2^{-1} \mathbf{x}_2 \\
&= \mathbf{K}_1 \mathbf{R}^\top \mathbf{K}_2^{-1} \mathbf{K}_2 \left[\mathbf{R}_2 \mid \mathbf{t}_2\right] \mathbf{X} \\
&= \mathbf{K}_1 \mathbf{R}^\top \left[\mathbf{R}_2 \mid \mathbf{t}_2\right] \mathbf{X} \\
&= \mathbf{K}_1 \left[\mathbf{R}_1 \mid \mathbf{t}_1\right] \mathbf{X} \\
&= \mathbf{x}_1
\end{aligned}$$

since $\left[\mathbf{R}_2 \mid \mathbf{t}_2\right] = \mathbf{R}\left[\mathbf{R}_1 \mid \mathbf{t}_1\right] \implies \mathbf{R}^\top \left[\mathbf{R}_2 \mid \mathbf{t}_2\right] = \left[\mathbf{R}_1 \mid \mathbf{t}_1\right]$.

Question 4. (Camera Calibration)

We are required to perform Camera Calibration with a set of 25 images of a chessboard pattern. A chessboard pattern of size 5×7 (given in [1]) was used for the calibration. The final set of images used for the camera calibration is given in Figure (1). The images are numbered 1 to 25 from top-left to bottom-right. The resolution of the images is 4032×3024.
The images were taken by placing the chessboard pattern on a flat surface and capturing the images from different angles and distances, capturing all degrees of freedom, esuring that the corners are clearly visible in each image. The code for this question is given in `camera-calibration.ipynb`.

1. We estimate the intrinsic camera parameters using the images. The following intrinsic matrix was obtained.

$$\mathbf{K} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 3127.88 & 0.0 & 1508.31 \\ 0.0 & 3122.10 & 2013.85 \\ 0 & 0 & 1 \end{bmatrix}$$

   inidicating that (rounded off to the nearest integer) the focal length in the x and y directions is $f_x \approx 3128$ and $f_y \approx 3122$ respectively. The principal point is $(c_x, c_y) \approx (1508, 2014)$. The skew parameter is $s = 0$.

2. We estimate the extrinsic camera parameters for each of the 25 images. A few of the results are given in Table 1. The figures are rounded off to one decimal place for brevity. The precise parameters for each image can be found in the notebook.

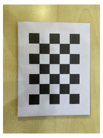
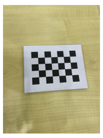
Image	Rotation Matrix	Translation Vector
	$\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$	$\begin{bmatrix} -1.2 \\ -2.5 \\ 9.9 \end{bmatrix}$
	$\begin{bmatrix} 0.9 & -0.1 & -0.3 \\ 0.1 & 0.9 & -0.4 \\ 0.3 & 0.4 & 0.8 \end{bmatrix}$	$\begin{bmatrix} -1.1 \\ -0.5 \\ 9.1 \end{bmatrix}$
	$\begin{bmatrix} 0.1 & -0.9 & -0.4 \\ 0.9 & 0.2 & -0.2 \\ 0.2 & -0.3 & 0.9 \end{bmatrix}$	$\begin{bmatrix} -13.2 \\ -23.5 \\ 48.4 \end{bmatrix}$

Table 1: Extrinsic camera parameters for the first three images

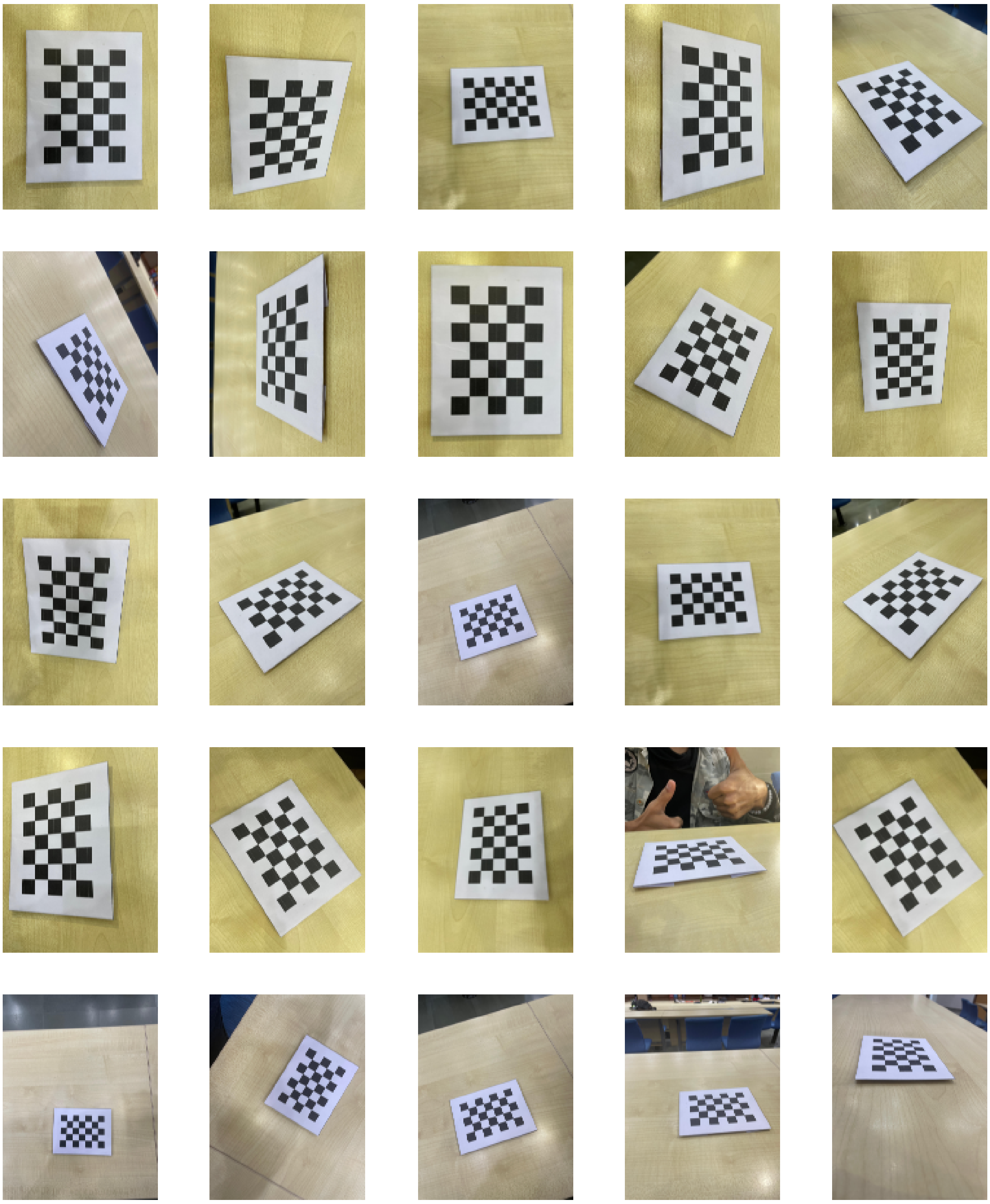


Figure 1: Chessboard pattern used for camera calibration

3. We estimate the distortion parameters, specifically the tangenial and radial distortion parameters using the images. The following distortion vector was obtained.

$$\mathbf{D} = \begin{bmatrix} k_1 & k_2 & p_1 & p_2 & k_3 \end{bmatrix} = \begin{bmatrix} 0.223 & -1.141 & -0.001 & -0.001 & 0.983 \end{bmatrix}$$

indicating that the radial distortion coefficients are $k_1 = 0.223$, $k_2 = -1.141$, and $k_3 = 0.983$. We then use these parameters to undisort 5 images and compare the results with the original images. The comparison us given in Figure (2).

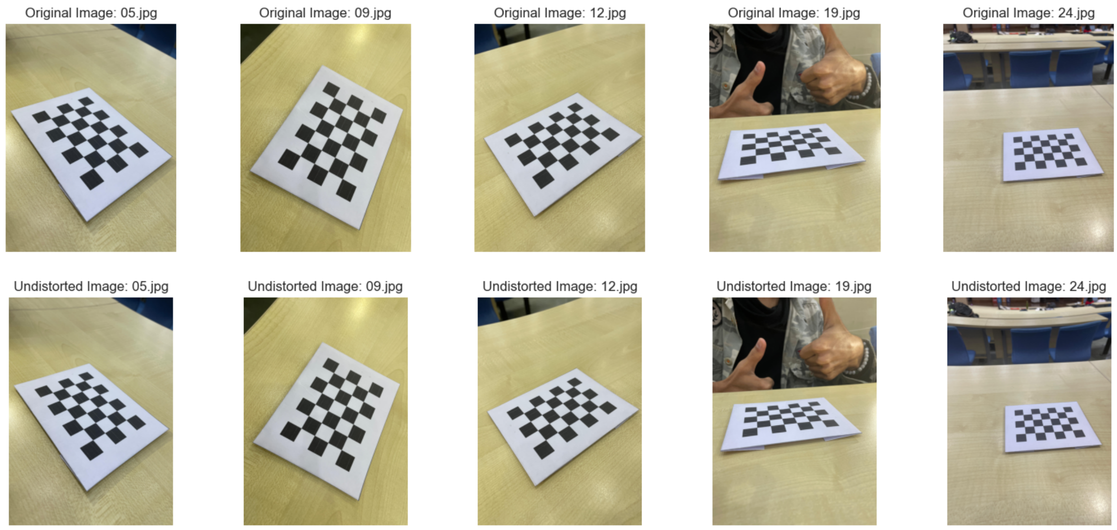


Figure 2: Comparison of 5 original and undisorted images

We can see that on the edges/corners of the undisorted images, the straight lines appear to be curved; for example, in image 24, the edges of the tables behind the chessboard appear to be curved inward, i.e. showing negative curvature. This is due to radial distortion. Similar effects are seen in images 9 and 12.

4. We compute the reprojection error using the estimated intrinsic and extrinsic parameters of each of the 25 images. A bar chart of the reprojection errors is given in Figure (3).

Figure 3: Reprojection error for each of the 25 images

The mean reprojection error is 0.40 pixels, with a standard deviation of 0.16 pixels.

5. We plot the figures of the chessboards showing the detected corners and the corners after reprojecting them using the estimated intrinsic and extrinsic parameters. The figures are given in Figure (4). In the

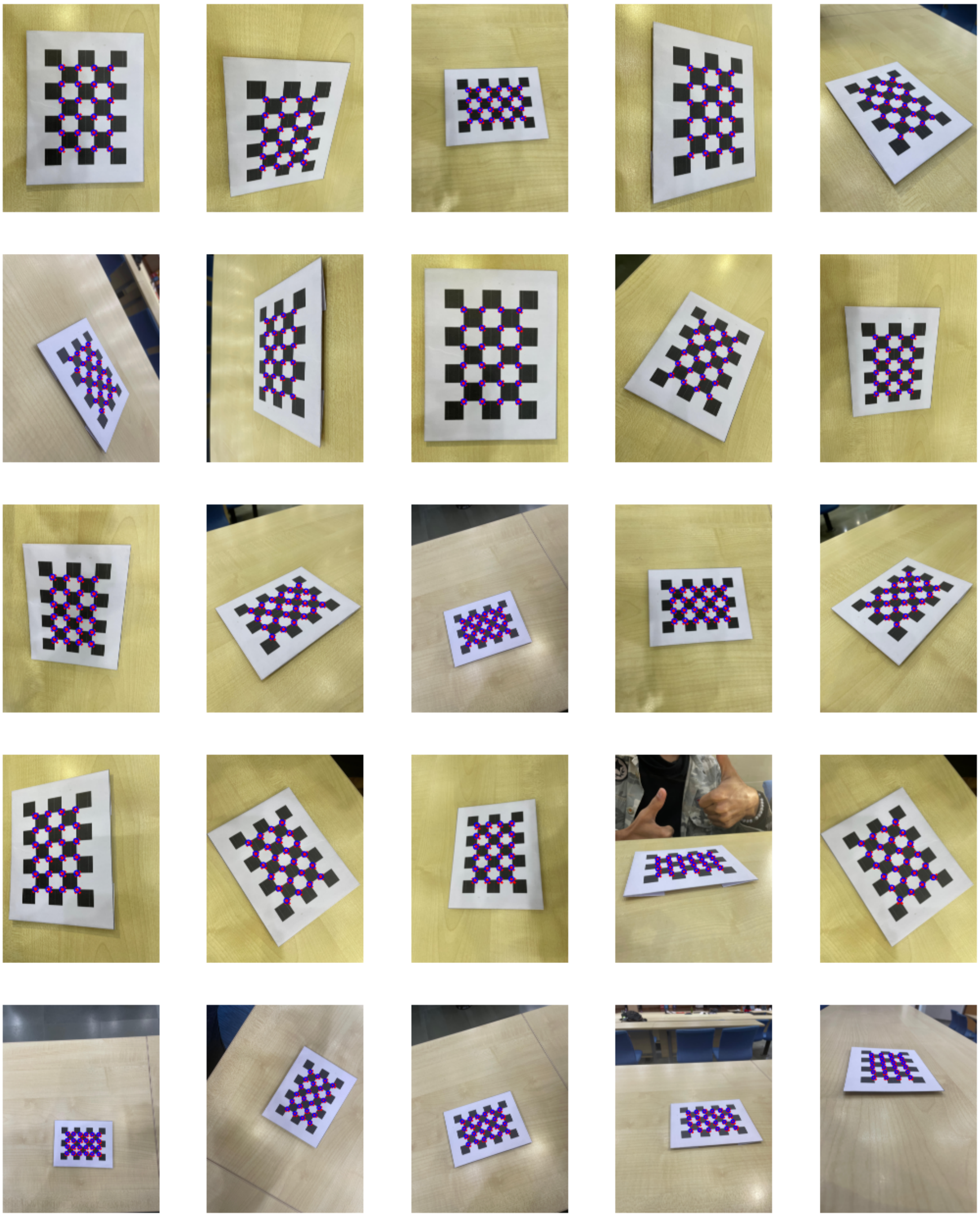


Figure 4: Detected and reprojected corners of the chessboard

figure, the red crosses are the detected corners, while the blue circles are the reprojected corners. The images are ordered as before. Reprojection error is computed as the average of the Euclidean distances between the detected and reprojected corners, i.e.

$$\mathbf{e}(\mathbf{c}, \mathbf{c}') = \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{c}_i - \mathbf{c}'_i\|_2$$

where $\mathbf{c}_i$ and $\mathbf{c}'_i$ are the observed and reprojected 2D coordinates of the i-th corner, and N is the number of corners.

6. We compute the chessboard plane normals $\hat{\mathbf{n}}_i^C$ in the camera coordinate frame for each image. The normals are given in Table 2. The figures are rounded off to two decimal places for brevity. The precise parameters for each image can be found in the notebook.

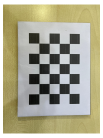
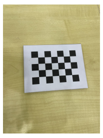
Image	Plane Normal
	$\begin{bmatrix} -0.03 & -0.01 & 0.99 \end{bmatrix}^\top$
	$\begin{bmatrix} -0.27 & -0.40 & 0.87 \end{bmatrix}^\top$
	$\begin{bmatrix} -0.42 & -0.16 & 0.89 \end{bmatrix}^\top$

Table 2: Plane normals for the first three images

As opposed to just numbers, we can visualize the normals on the chessboard. The figures are given in Figure (5). The normals are shown as red lines on the chessboard, while the green diagonals indicate the chessboard plane.

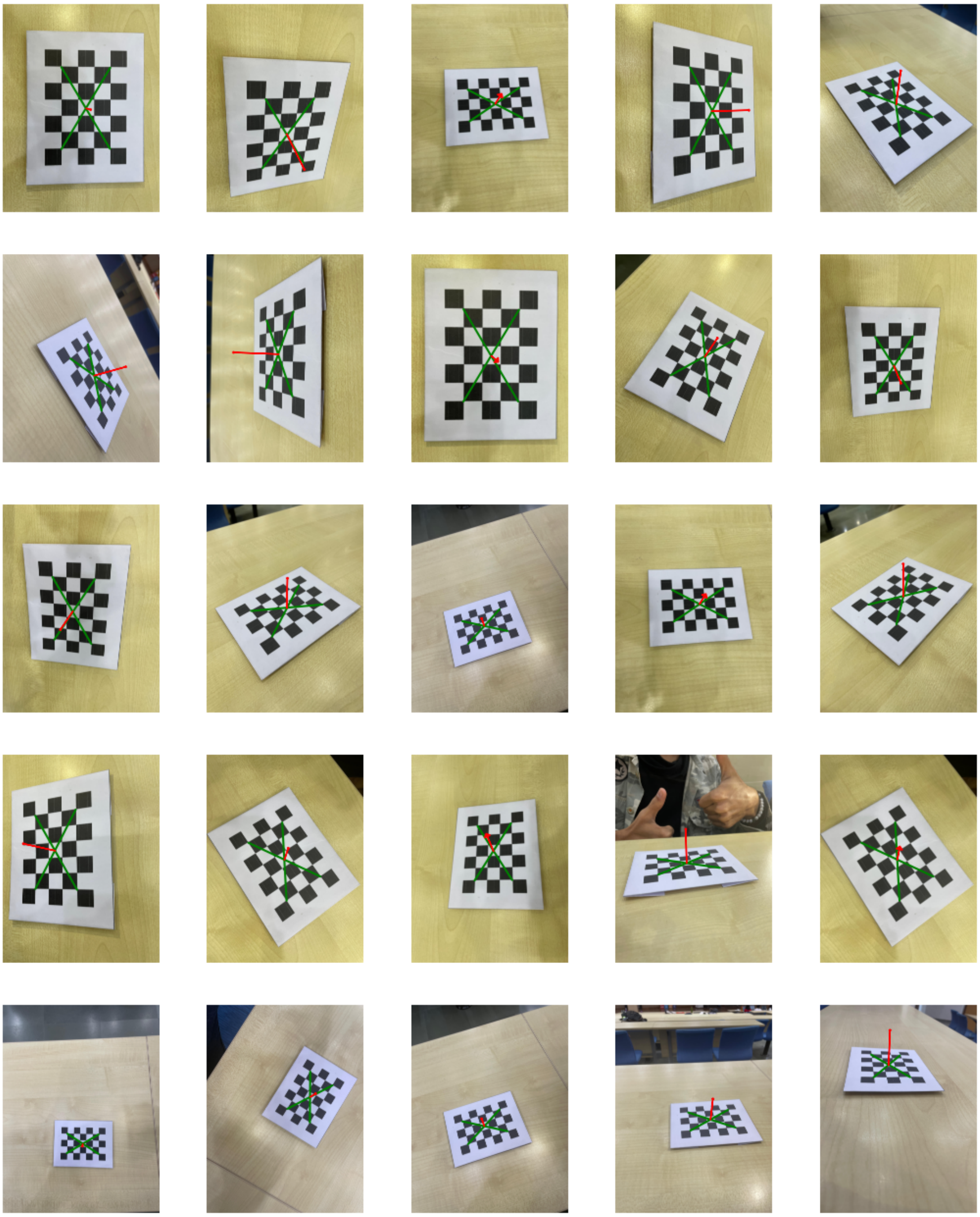


Figure 5: Plane normals for each of the 25 images

Question 5. (Camera-LIDAR Cross-Calibration)

We are required to perform Camera-LIDAR Cross-Calibration with the given set of images and their corresponding LIDAR scans. From the given dataset, we choose a set of 25 images to perform the cross-calibration, which are given in Figure (6). The code for this question is given in `LIDAR-cross-calibration.ipynb`.

Figure 6: Images used for camera-LIDAR cross-calibration

1. We first compute the chessboard plane normals $\hat{\mathbf{n}}_i^L$ in in the LIDAR coordinate frame using the points given in the `.pcd` files and their corresponding offsets. We calculate the normals using singular value decomposition (SVD) of the matrix of points as follows. For the i^{th} scan, let

$$X_{N\times 3}^{(i)} = \begin{bmatrix} \mathbf{x}_1^{(i)} & \mathbf{x}_2^{(i)} & \cdots & \mathbf{x}_N^{i} \end{bmatrix}^\top$$

   be the matrix of points in the LIDAR coordinate frame, where each $\mathbf{x}_j^{(i)}$ is a 3D point and N is the number of points. Then, the normal $\hat{\mathbf{n}}_i^L$ to the plane of the points is given by the rightmost column of the matrix V where

$$X^{(i)} - \bar{\mathbf{x}}^{(i)} = U\Sigma V^\top$$

   is the singular value decomposition of the matrix of centered points $X^{(i)} - \bar{\mathbf{x}}^{(i)}$, where $\bar{\mathbf{x}}^{(i)}$ is the centroid or mean of the points. The offset is simply

$$\theta_i = \hat{\mathbf{n}}_i^L \cdot \bar{\mathbf{x}}^{(i)}$$

   which is the scalar in the equation of the plane. In case the offset is negative, we negate the normal and the offset. The normals and offsets for the first three scans are given in Table 3. The figures are rounded off to two decimal places for brevity. The precise parameters for each scan can be found in the notebook.

Scan	Plane Normal	Offset
	$\begin{bmatrix} 0.64 & -0.77 & 0.1 \end{bmatrix}^\top$	5.0

	$\begin{bmatrix}0.7 & -0.7 & 0.12\end{bmatrix}^\top$	4.72
	$\begin{bmatrix}0.94 & -0.23 & 0.26\end{bmatrix}^\top$	5.2

Table 3: Plane normals and offsets for the first three LIDAR scans

2. We estimate the LIDAR-to-Camera transformation matrix. Using the closed form solution given in the thesis [2], if

$$\mathbf{n}_C = \begin{bmatrix}\hat{\mathbf{n}}_1^{(C)} \\ \hat{\mathbf{n}}_2^{(C)} \\ \vdots \\ \hat{\mathbf{n}}_N^{(C)}\end{bmatrix}_{N\times 3} \qquad \mathbf{n}_L = \begin{bmatrix}\hat{\mathbf{n}}_1^{(L)} \\ \hat{\mathbf{n}}_2^{(L)} \\ \vdots \\ \hat{\mathbf{n}}_N^{(L)}\end{bmatrix}_{N\times 3} \qquad \mathbf{d}_C = \begin{bmatrix}d_1^{(C)} \\ d_2^{(C)} \\ \vdots \\ d_N^{(C)}\end{bmatrix}_{N\times 1} \qquad \mathbf{d}_L = \begin{bmatrix}d_1^{(L)} \\ d_2^{(L)} \\ \vdots \\ d_N^{(L)}\end{bmatrix}_{N\times 1}$$

then

$$\begin{aligned} {}^{C}t_L &= (\mathbf{n}_C^\top \mathbf{n}_C)^{-1}\mathbf{n}_C^\top (d_C - d_L) \\ {}^{C}\mathbf{R}_L &= VU^\top \quad \text{if} \quad \mathbf{n}_L^\top \mathbf{n}_C = U\Sigma V^\top \end{aligned}$$

where $\mathbf{n}_C$ and $\mathbf{n}_L$ are the matrices of normals in the camera and LIDAR coordinate frames, N is the number of points, $\mathbf{d}_C$ and $\mathbf{d}_L$ are the vectors of offsets in the camera and LIDAR coordinate frames, and U, Σ, and V are the matrices obtained from the singular value decomposition of $\mathbf{n}_L^\top \mathbf{n}_C$. Then, the transformation matrix is simply

$${}^{C}\tilde{\mathbf{T}}_L = \begin{bmatrix}{}^{C}\mathbf{R}_L \mid {}^{C}t_L\end{bmatrix}_{4\times 3}$$

The above equations suggest that the translation vector ${}^{C}t_L$ is obtained as the solution to the equation

$$\mathbf{n}_C \cdot {}^{C}t_L = \mathbf{d}_C - \mathbf{d}_L$$

which makes sense, as the dot product of the normal and the translation vector gives the distance of the plane from the origin. The rotation matrix ${}^{C}\mathbf{R}_L$ is obtained as the product of the orthogonal matrices V and $U^\top$ from the SVD of $\mathbf{n}_L^\top \mathbf{n}_C$. This also aligns with intuition, since the SVD of a linear transform (in our case $\mathbf{n}_L^\top \mathbf{n}_C$) decomposes it into a rotation/reflection $V^\top$, followed by a scaling Σ, and another rotation U. So the rotation matrix is obtained as the composition of both rotations.

3. The above closed form solution was implemented in the notebook to estimate the transformation matrix. We use it to obtain the following transformation matrix (rounded off to two decimal places for brevity).

$${}^{C}\tilde{\mathbf{T}}_L = \begin{bmatrix} -0.18 & -0.98 & 0.0 & 0.15 \\ 0.02 & 0.0 & -0.99 & -0.41 \\ 0.98 & -0.18 & 0.02 & -0.6 \end{bmatrix}$$

It can be noted that the determinant of the rotation matrix (first three columns) has a determinant +1, as can be seen in the code in the notebook.

4. We use the estimated transformation matrix ${}^{C}\tilde{\mathbf{T}}_L$ to map the LIDAR points to the camera coordinate frame, and then project them to the image plane using the camera's (given) intrinsic parameters and the (given) extrinsic parameters for each image. The figures are given in Figure (7). The points were transformed from the LIDAR coordinates to image coordinates using the following equation

$$\mathbf{X}^{(C)} = \mathbf{K}\ {}^{C}\tilde{\mathbf{T}}_L \left(\mathbf{X}^{(L)}\right)^\top$$

where $\mathbf{X}^{(C)}$ is the $3 \times N$ matrix of points in the homogeneous camera coordantes, $\mathbf{K}$ is the intrinsic camera matrix, and $\mathbf{X}^{(L)}$ is the $4 \times N$ matrix of points in the homogeneous LIDAR coordinates.
It can be easily seen that not all points are within the checkerboard pattern's boundary. For example, in images 1 and 9, some of the projected points are to the right of the checkerboard. However, the points are close by. This could be due to the error in estimating the transformation matrix.

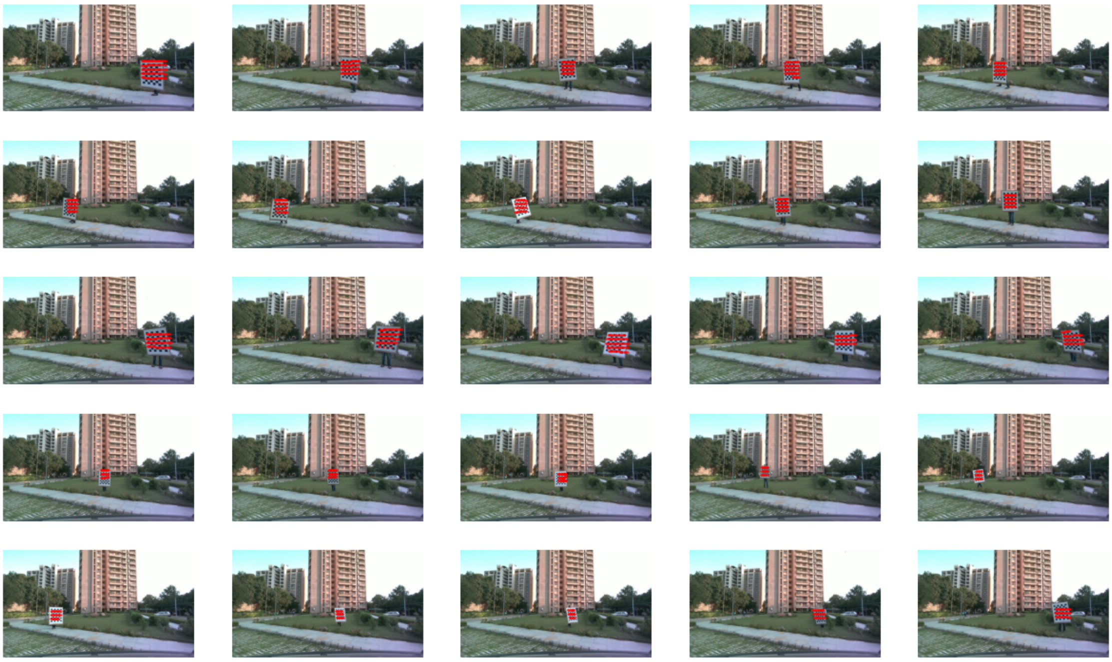


Figure 7: LIDAR points projected to the image plane

5. We plot the normals $\hat{\mathbf{n}}_i^L$, $\hat{\mathbf{n}}_i^C$, and ${}^C\mathbf{R}_L\hat{\mathbf{n}}_i^L$ for 5 selected images. The figure is given in Figure (8). In the figure, the blue normals are the LIDAR normals $\hat{\mathbf{n}}_i^L$, the green normals are the camera normals $\hat{\mathbf{n}}_i^C$, and the orange normals are the transformed LIDAR normals ${}^C\mathbf{R}_L\hat{\mathbf{n}}_i^L$.

Figure 8: Normal vectors for 5 selected images

The transformed LIDAR normals are close to the camera normals, whereas the LIDAR normals are not. This is expected, as the transformation matrix maps LIDAR points (and therefore vectors) to the camera coordinate frame.
We calculate the cosine distance (normalized to $[0, 1]$) between the camera normals and the transformed LIDAR normals. The distribution of these errors is given in Figure (9).
The mean cosine distance is 0.0003, with a standard deviation of 0.0014.

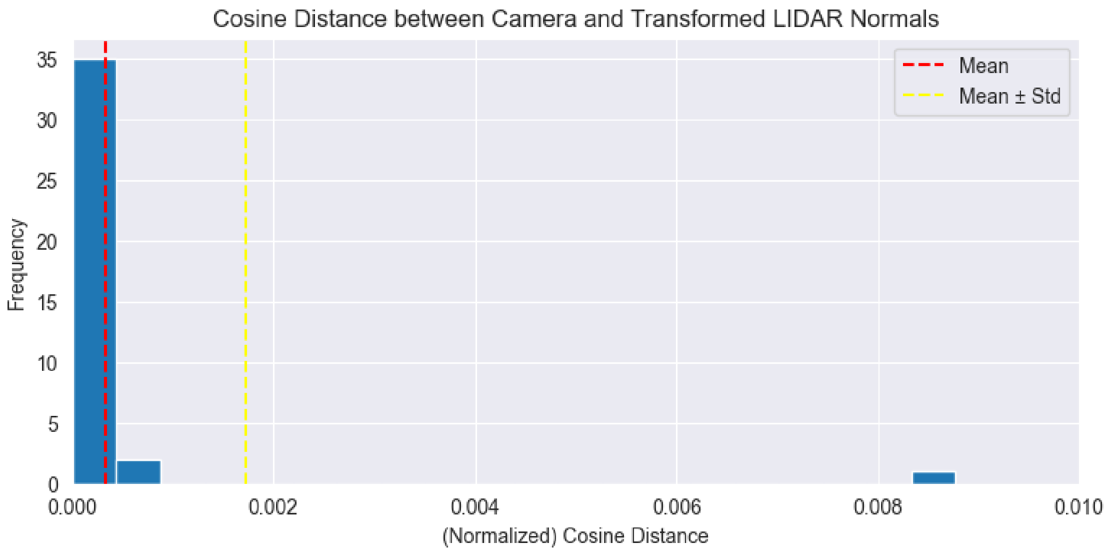


Figure 9: Cosine distance between camera and transformed LIDAR normals

References

1. Camera Calibration using OpenCV
2. Ranjith Unnikrishnan, Fast Extrinsic Calibration of a Laser Rangefinder to a Camera, 2005