MOBILE ROBOTICS

Assignment 1 Report

Transformations, Camera Modelling and DLT

AadilMehdi Sanchawala (20171043)
Rahul Sajnani (20171056)

Question 1 (q1_script.py)

The Lidar's Frame is defined as follows,

- X axis points forward
- Y axis points left
- Z axis points upward

The Camera's Frame is defined as follows,

- X axis points right
- Y axis points downward
- Z axis points forward

Thus the relative orientation of the Camera with respect to the world can be defined as follows,

$$\mathbf{R_z(-90) * R_x(-90)}$$

The camera is placed 8 cm below, 6 cm left, and 27 cm in front of the LiDAR's center.

Thus the translation can be given as

$$\mathbf{T = [0.27, 0.06, -0.08]}$$

Thus the transformation from the camera's frame of reference to the world's frame of reference is given as,

$$\mathbf{M = T * R_z(-90) * R_x(-90)}$$

This transformation M maps the points in the camera's frame of reference to the world's frame of reference as follows,

$$\mathbf{X_w = M * X_c}$$

However, we need to transform the world points into the camera's frame of reference thus we multiply the above equation with $\mathbf{M^{-1}}$ getting the following equation,

$$\mathbf{X_c = M^{-1} * X_w}$$

Let **L = M^{-1},**

Thus **L** can be given as follows,

$$\mathbf{L = R_x(90) * R_z(90) * (-T)}$$

Thus the world points in camera's frame of reference are given by,

$$\mathbf{X_c = L * X_w}$$

$$\mathbf{x = K * X_c}$$

Euler angles is given as **(90, 0, 90)**

Rx(90) =

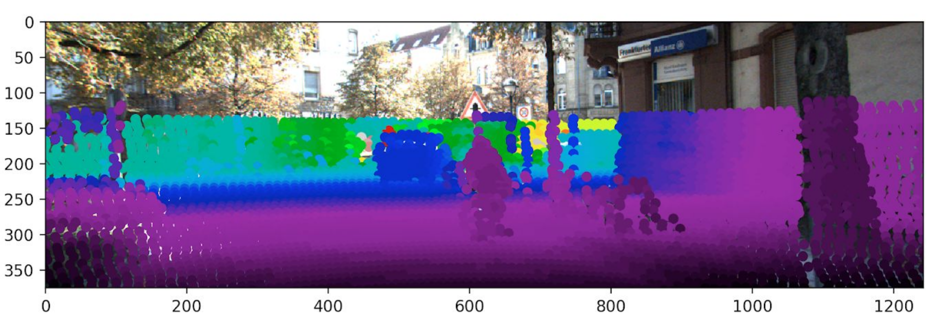
1	0	0
0	0	-1
0	1	0

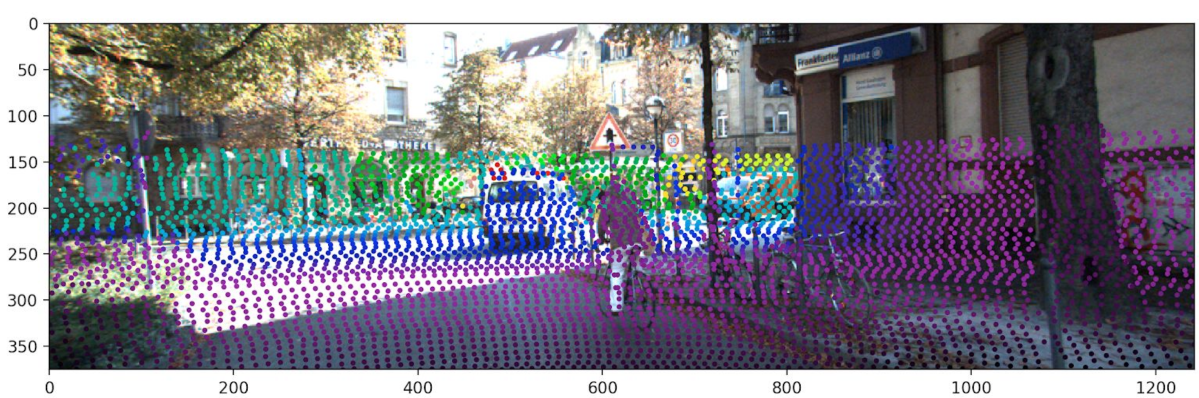
Rz(90) =

0	-1	0
1	0	0
0	0	1

(-T) =

1	0	0	-0.27
0	1	0	-0.06
0	0	1	0.08





Question 2 (q2_script.py)

Approach

The information provided in the question gives us the height of the camera from the ground. We also know that the ground is perpendicular to the Y axis.

With this information, we back project the ray from the image coordinates to the world. The equation of this ray is given as follows,

$$\overline{P} = K^{-1}x$$

where x is the homogeneous coordinate of the image pixel corresponding to the world origin. We have chosen the rear left tire to be the origin of our world coordinate system and $\overline{x} = [835, 305, 1]^T$.

Now we chose a normal vector pointing to the ground and perpendicular to the ground as $\overline{N} = [0, 1, 0]^T$.

Taking dot product of $\overline{N}$ and $\overline{P}$ we get,

$$\overline{N} \bullet \overline{K^{-1}x} = \|N\| * \left\|K^{-1}x\right\| * cos(\theta),$$

where θ is the angle between $\overline{N}$ and $\overline{P}$.

To find the exact magnitude of the world origin in terms of camera's frame we need to calculate $\frac{H}{cos(\theta)}$, where $H = 1.65m$

Thus the world origin in camera's frame of reference is

$$\frac{H}{cos(\theta)} = H * \frac{\left\|K^{-1}x\right\|}{\overline{N} \bullet \overline{K^{-1}x}}$$

Thus, we obtain the world origin in camera's frame of reference. **N is a unit vector.**

Now in order to compute the coordinates of the rest of the points in the camera's frame, we add the dimensions of the car accordingly to the coordinates of the world origin in camera's frame.

To obtain the image coordinates of these points we multiply them with the camera calibration matrix K.

Thus we obtain the endpoints of the car in the image and plot the edges accordingly.

To account for the 5˚ rotation along the Y axis, we rotate the world points before transforming them in the camera's frame of reference.

Question 3 (q3_script.py)

We first found the homography matrix corresponding to each point using DLT but we ignore the Z value as we do not have any depth info (all the points are lying on the same plane). After finding the homography matrix(3x3) we try to estimate its pose. To do this we first multiply H with inv(K) [inv(K)*H] this gives us a 3x3 matrix. Now we have to decompose this 3x3 matrix to find rotation 1 ,rotation 2 and translation of the camera(extrinsic parameters).

H1 = inv(K)*H

We first divide H1 by the first column vector of H1. Then we take the first two columns of H1 and their cross product as r1, r2 and r3 respectively.

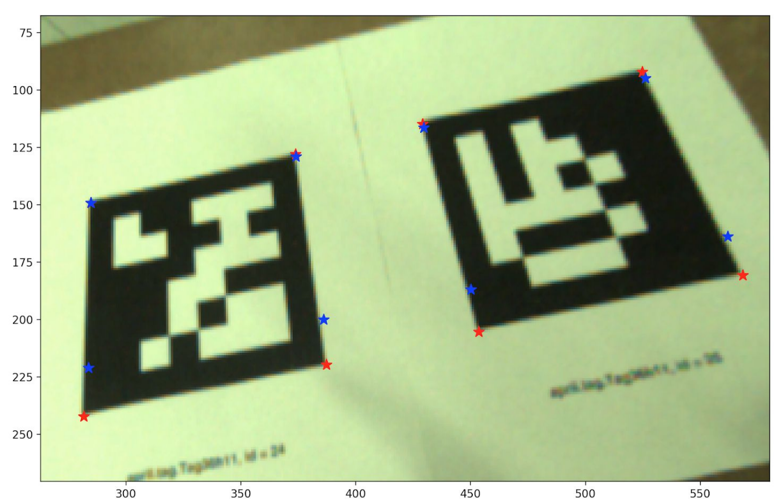
R = [r1, r2, r3]

We SVD decompose the R matrix and force eigenvalue diagonals to be [1, 1, det(UV^T)]Here U and V are the left and right eigenvectors of R obtained using SVD.

After remultiplying the decomposed R matrix we choose the first 2 columns of R as r1 and r2 of camera and last column of H1 as T.

Decomposed H = K* [r1,r2,T]

Link used for



decomposition: https://fling.seas.upenn.edu/~cis390/dynamic/slides/CIS390_Lecture12.pdf

Homography Matrix

[[6.80262999e-01 -3.92798233e-01 3.13972339e-01]

[-2.00053723e-01 4.66175179e-01 1.64406319e-01]

[-1.78910003e-04 -1.30147121e-03 1.10297892e-03]]

Decomposed [r1, r2, T]

[[0.97604007 -0.1908006 -0.10459882]

[0.10131016 0.82392071 -0.55757593]

[-0.11746007 -0.13824321 0.58632706]]