Robot Localization and Mapping - 16833
Carnegie Mellon University
Fall 2023
Sai Sravan Yarlagadda
andrew id: saisravy

Homework #4: Dense SLAM with Point-based Fusion

Sai Sravan Yarlagadda
andrew id: saisravy

1 Projective Data Association

In this question we had to write down conditions u, v and d must satisfy to setup valid correspondences. The conditions taken into consideration are:

$$w = \text{target vertex map width}$$
$$h = \text{target vertex map height}$$
$$w > u \geq 0$$
$$h > v \geq 0$$
$$d \geq 0$$

The code for this is written is icp.py

After obtaining the correspondences q from vertex map and the corresponding normal n_q in the normal map we need to additionally filter them based on distance threshold . This is necessary to remove the source points that are projected very far from the target points. Hence we can set a distance threshold based on the L2 norm of difference between source and the target points.
The condition that needs to be satisfied is: $|p - q| < d_{\text{thr}}$
The code for this is written in icp.py

2 Linearization

$$(R \times \begin{bmatrix} P_{\text{ix}} \\ P_{\text{iy}} \\ P_{\text{iz}} \end{bmatrix} + \begin{bmatrix} t_{\text{x}} \\ t_{\text{y}} \\ t_{\text{z}} \end{bmatrix} - \begin{bmatrix} q_{\text{ix}} \\ q_{\text{iy}} \\ q_{\text{iz}} \end{bmatrix}) \times \begin{bmatrix} n_{\text{ix}} & n_{\text{iy}} & n_{\text{iz}} \end{bmatrix}$$

R is the rotation matrix and is a 3×3 matrix
t is the translation matrix and is a 3×1matrix

By considering the small angle assumption that is:

$$R = \begin{bmatrix} 1 & -\gamma & \beta \\ \gamma & 1 & -\alpha \\ -\beta & \alpha & 1 \end{bmatrix}$$

Substituting above mentioned R in the equation above that we get:

$$\sum_i r_i(R,t) = n_{\text{ix}} \times (P_{\text{ix}} - \gamma \times P_{\text{iy}} + \beta \times P_{\text{iz}} + t_x + q_{\text{ix}}) + n_{\text{iy}} \times (\gamma \times P_{\text{ix}} + P_{\text{iy}} - \alpha \times P_{\text{iz}} + t_y + q_{\text{iy}} + n_{\text{iz}} \times (-\beta \times P_{\text{ix}} + \alpha \times P_{\text{iy}} + P_{\text{iz}} + t_z + q_{\text{iz}})$$

By rearranging the above equation in the form of:

$$r_i(\alpha, \beta, \gamma, t_x, t_y, t_z) = A_i \times \begin{bmatrix} \alpha \\ \beta \\ \gamma \\ t_x \\ t_y \\ t_z \end{bmatrix} + b_i$$

By rearranging the equation above this, we can get our A_i matrix and b_i matrix.

$$A_i = \begin{bmatrix} n_{\text{iz}}P_{\text{iy}} - n_{\text{iy}}P_{\text{iz}} & n_{\text{ix}}P_{\text{iz}} - n_{\text{iz}}P_{\text{ix}} & n_{\text{iy}}P_{\text{ix}} - n_{\text{ix}}P_{\text{iy}} & n_{\text{ix}} & n_{\text{iy}} & n_{\text{iz}} \end{bmatrix}$$

$$\text{and } b_i = \sum_i n_i \times (d_i - P_i)$$

From the above equation A_i is a 1×6 matrix and b_i is a scalar.

3 Optimization

Using the equations above I have build the linear model and the solved the linear least squares equation using numpy's QR library.
My implementation can be seen in the build linear system function and the corresponding solve function from icp.py.

Figure 1 and 2 are the results with default source and targets(frame 10, 50)

Figure 1: Point cloud before registration

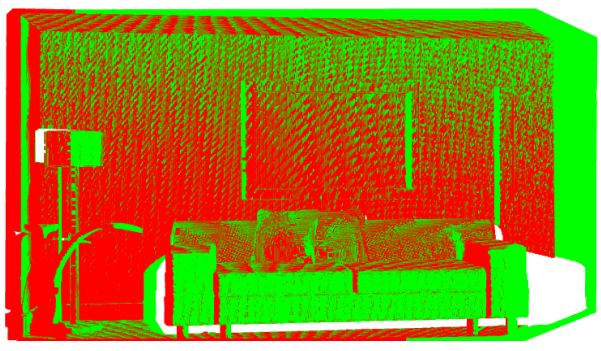

Figure 2: Point cloud after registration

Figure 3: Point cloud before registration

Figure 4: Point cloud after registration

Figure 3 and Figure 4 shows the point cloud before and after registration between frames 10 and 100. The poor outcome is attributed to the substantial variance between frames, which can result in inadequate correspondences

4 Point-based Fusion

The weight average of the positions in terms of p, q, R_w^c, t_w^c, color (c) and w is:

$$q = \frac{w \times q + (R_w^c \times p + t_w^c)}{1+w}$$

$$n_q = \frac{w \times n_q + (R_w^c \times n_p}{1+w}$$

$$c_q = \frac{c_q \times w}{1+w}$$

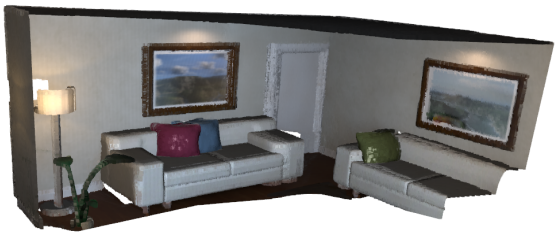

Figure 5: Map - after fusion

Figure 6: Normal Map- after fusion

In Figures 5 and 6, I showed the Map and the normal map, respectively after running fusion.py. The final updated number of entries is 147067. The compression ratio obtained is 0.1122.

Question Which is the source and which is the target, for ICP between the map and the input RGBD-frame? Can we swap their roles, why or why not?
Answer In ICP, the points we are trying to align on to the map are the source points and the points we are trying to align the source points with are the target points. We cannot swap the source and target points as if we do this we do not have estimate to where the target points are originally onto the map.

Now after running main.py, I have shown my visualizations below in figures 7 and 8 respectively.

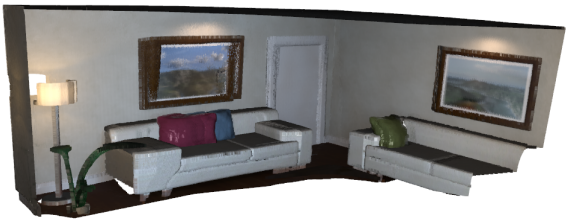

Figure 7: Final Visualization- main.py

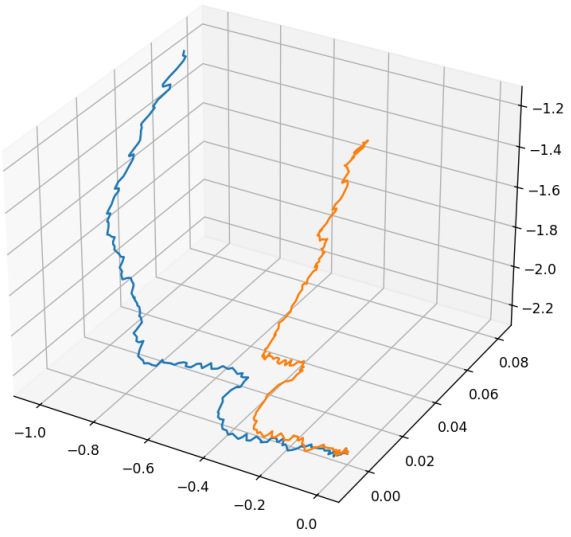

Figure 8: Estimated Trajectory vs Ground Truth