Homework 4 16833: Robot Localization and Mapping

Harsh Dhruva
Dense SLAM with Point-based Fusion

April 25, 2021

---

1 Overview

2 Iterative Closest Point

2.1 Projective data association

Question
Conditions for u, v, and d to satisfy for a valid correspondence.

$$0 \leq \text{u} < \text{W}$$

$$0 \leq \text{v} < \text{H}$$

$$0 \leq \text{d}$$

Implemented first filter in code

Question

Since, p and q are projective nearest neighbors $|p - q| < d_{thr}$ is necessary. This is because we need to ensure that the points are within a distance threshold in the camera viewpoint direction, as even though p and q would be close to each other in 2-dimensional space, we need to ensure they are nearest neighbors in 3-dimensional space.

Implemented second filter in code

2.2 Linearization

Question

$$\sum_{i\in\Omega} r_i^2(\delta R, \delta t) = \left\| n_{q_i}^\top \Big((\delta R) p_i' + \delta t - q_i \Big) \right\|^2$$

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

$$n_{q_i} = \begin{bmatrix} n_{q_i x} \\ n_{q_i y} \\ n_{q_i z} \end{bmatrix} \; p_i' = \begin{bmatrix} p_{ix}' \\ p_{iy}' \\ p_{iz}' \end{bmatrix} \; q_i = \begin{bmatrix} q_{iy} \\ q_{iy} \\ q_{iz} \end{bmatrix}$$

$$A_i = \begin{bmatrix} n_{q_i z} p_{iy}' - n_{q_i y} p_{iz}' & n_{q_i x} p_{iz}' - n_{q_i z} p_{ix}' & n_{q_i y} p_{ix}' - n_{q_i x} p_{iy}' & n_{q_i x} & n_{q_i y} & n_{q_i z} \end{bmatrix}_i$$

$$b_i = n_{q_i x}(p_{ix}' - q_{ix}) + n_{q_i y}(p_{iy}' - q_{iy}) + n_{q_i z}(p_{iz}' - q_{iz})$$

2.3 Optimization

Question

We can write the system in the form of $Ax = b$ where x $= \begin{bmatrix} \alpha \\ \beta \\ \gamma \\ t_x \\ t_y \\ t_z \end{bmatrix}$

and A $= \begin{bmatrix} n_{q_1z}p'_{1y} - n_{q_1y}p'_{1z} & n_{q_1x}p'_{1z} - n_{q_1z}p'_{1x} & n_{q_1y}p'_{1x} - n_{q_1x}p'_{1y} & n_{q_1x} & n_{q_1y} & n_{q_1z} \\ n_{q_2z}p'_{2y} - n_{q_2y}p'_{2z} & n_{q_2x}p'_{2z} - n_{q_2z}p'_{2x} & n_{q_2y}p'_{2x} - n_{q_2x}p'_{2y} & n_{q_2x} & n_{q_2y} & n_{q_2z} \\ n_{q_3z}p'_{3y} - n_{q_3y}p'_{3z} & n_{q_3x}p'_{3z} - n_{q_3z}p'_{3x} & n_{q_3y}p'_{3x} - n_{q_3x}p'_{3y} & n_{q_3x} & n_{q_3y} & n_{q_3z} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \end{bmatrix}$

and b $= \begin{bmatrix} n_{q_1x}(q_{1x} - p'_{1x}) + n_{q_1y}(q_{1y} - p'_{1y}) + n_{q_1z}(q_{1z} - p'_{1z}) \\ n_{q_2x}(q_{2x} - p'_{2x}) + n_{q_2y}(q_{2y} - p'_{2y}) + n_{q_2z}(q_{2z} - p'_{2z}) \\ n_{q_3x}(q_{3x} - p'_{3x}) + n_{q_3y}(q_{3y} - p'_{3y}) + n_{q_3z}(q_{3z} - p'_{3z}) \\ . \\ . \\ . \end{bmatrix}$ and

x can be solved by solving $A^TAx = A^Tb$ to get $x = (A^TA)^{-1}A^Tb$

Note b matrix is different from b_i in 2.2, as sign is flipped to write in terms of $Ax = b$

Question

Frame 10 and 50 visualization

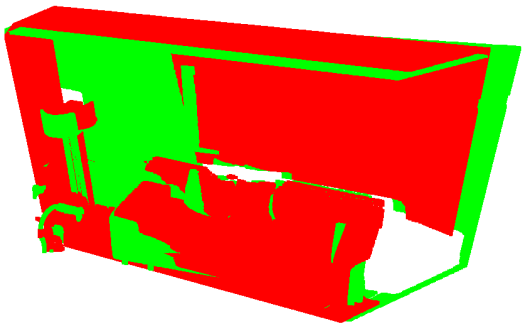

Figure 1: Before Alignment

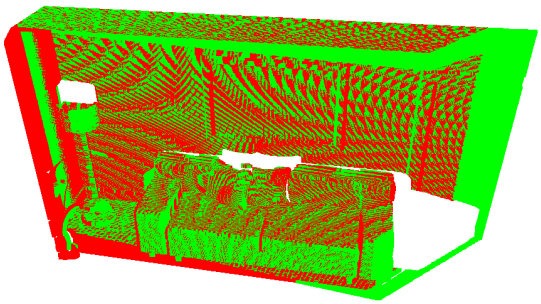

Figure 2: After Alignment

Frame 10 and 100 visualization

For alignment success, 80 iterations were needed. As the difference between frames is large, there is large misalignment initially, which therefore requires larger iterations. However we can see that the alignment is no completely satisfactory. During the alignment, it is possible to converge to solutions with local minima, like the pillows on the couch which align with different pillows.

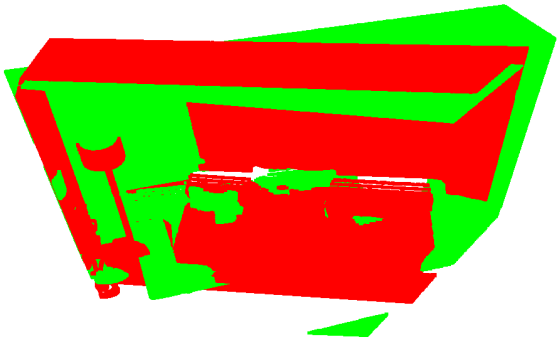

Figure 3: Before Alignment

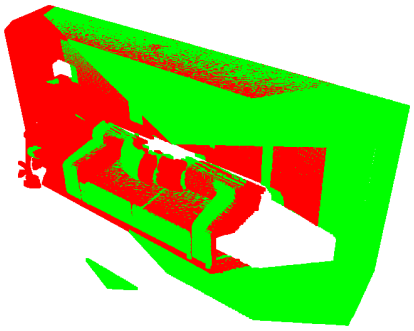

Figure 4: After Alignment

3 Point-based Fusion

3.1 Filter

Implemented in code

3.2 Merge

Question

$$p \leftarrow \frac{wp + (R_c^w q + t_c^w)}{w+1}$$
$$n_p \leftarrow \frac{wn_p + (R_c^w n_q + t_c^w)}{w+1}$$

Normalize normals after update

3.3 Addition

Implemented in code

3.4 Results

Figure 5: Fusion with ground truth poses

Figure 6: Normal map

Number of Points in the map = 1322141 , Compression Ratio = 0.0867689

4 The dense SLAM system

Question

The source is the maintained map and the target is the input RGBD-frame.

The points in the maintained map are projected to the input RGBD-frame vertex map, and filtered to then solve the linear system find the transformation. We cannot go the other way round in our case, since we need to find the transformation. As the map is increasing in size as points are added, projecting the RGBD-frame to the map could potentially lead to wrong transformations if the points get stuck in local minima. However, theoretically, if the the source and target are flipped it should work for the initial iterations, as the map size is small.

Question

Visualization Estimated Trajectory

Figure 7: Fusion with estimated poses

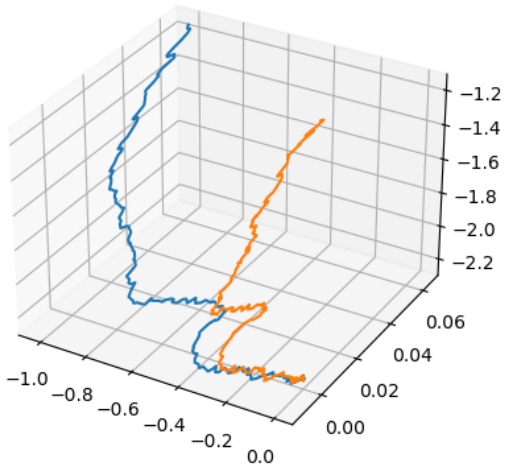

Figure 8: Estimated Trajectory vs Ground Truth