16-833 HW4:
Dense SLAM with Point-based Fusion

Yash Jain (Andrew ID - yashjain)

1 Iterative Closest Point

1.1 Projective data association

1.1.1 Question 1

The conditions u, v, d must satisfy are as follows-

$$0 \leq u < W \tag{1}$$

$$0 \leq v < H \tag{2}$$

$$0 \leq d \tag{3}$$

The points need to satisfy these conditions to be inside the vertex map.

1.1.2 Question 2

The second filter implements the following condition-

$$|p - q| < d_{thr} \tag{4}$$

This step is necessary because we wish to fuse data points within close proximity to reduce redundancy and drift.

1.2 Linearization

We obtain the linearized model as follows-

$$n_{qi}^T * ((\delta R)p_i^{'} + \delta t - q_i)) \tag{5}$$

where,

$$\delta R = \begin{bmatrix} 1 & -\gamma & \beta \\ \gamma & 1 & -\alpha \\ -\beta & \alpha & 1 \end{bmatrix} p_i^{'} = \begin{bmatrix} p_x^{'} \\ p_y^{'} \\ p_z^{'} \end{bmatrix} \delta t = \begin{bmatrix} \delta t_x \\ \delta t_y \\ \delta t_z \end{bmatrix} q_i = \begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix} \tag{6}$$

from eq(5) and eq(6) we get-

$$\begin{bmatrix} -n_2 p_z^{'} + n_3 p_y^{'} & n_1 p_z^{'} - n_3 p_x^{'} & -n_1 p_y^{'} + n_2 p_x^{'} & n_1 & n_2 & n_3 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \gamma \\ t_x \\ t_y \\ t_z \end{bmatrix} + \begin{bmatrix} n_1(p_x^{'} - q_x) + n_2(p_y^{'} - q_y) + n_3(p_z^{'} - q_z) \end{bmatrix} \tag{7}$$

where,

$$A_i = \begin{bmatrix} -n_2 p_z^{'} + n_3 p_y^{'} & n_1 p_z^{'} - n_3 p_x^{'} & -n_1 p_y^{'} + n_2 p_x^{'} & n_1 & n_2 & n_3 \end{bmatrix} \tag{8}$$

$$b_i = \begin{bmatrix} n_1(p_x^{'} - q_x) + n_2(p_y^{'} - q_y) + n_3(p_z^{'} - q_z) \end{bmatrix} \tag{9}$$

1.3 Optimization

1.3.1 Question 1

I used LU decomposition to solve the linear system derived in the above section. Please refer code for more details.

1.3.2 Question 2

Frame 10 and 50-

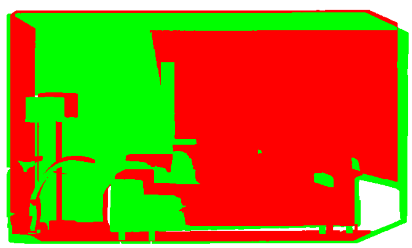

Figure 1: Before ICP

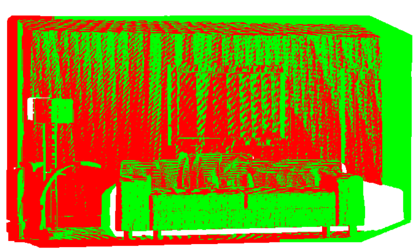

Figure 2: After ICP

Frame 10 and 100-

Figure 3: Before ICP

Figure 4: After ICP

The algorithm fails in the case of frames 10 and 100 because the frames are quite far apart and the algorithm needs more iterations to achieve convergence.

2 Point-based Fusion

2.1 Filter

Implemented in code.

2.2 Merge

Points-

$$p = \frac{w.p + q}{w + 1} = \frac{w.p + R_c^w.p + t}{w + 1} \tag{10}$$

Normals-

$$n_p = \frac{w.n_p + n_q}{w + 1} = \frac{w.n_p + R_c^w.n_p + t}{w + 1} \tag{11}$$

$$n_p = \frac{n_p}{||n_p||_2} \tag{12}$$

Colors-

$$c = \frac{w.c + c}{w + 1} \tag{13}$$

2.3 Addition

Implemented in code.

2.4 Results

Total Points - 14736705
Compression ratio-

$$cr = \frac{totalpoints}{W.H.frames} = 0.22574 \tag{14}$$

Figure 5: Visualization

Figure 6: Normal Map

3 The dense SLAM system

3.1 Question 1

For the ICP, the RGBD-frame is the source and the map is the target. We cannot swap their roles as the target is obtained after performing a number of transformations in a particular sequence, and simply reversing the roles would not lead to the desired result.

3.2 Question 2

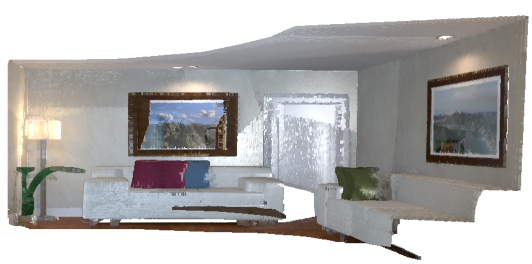

Figure 7: Visualization

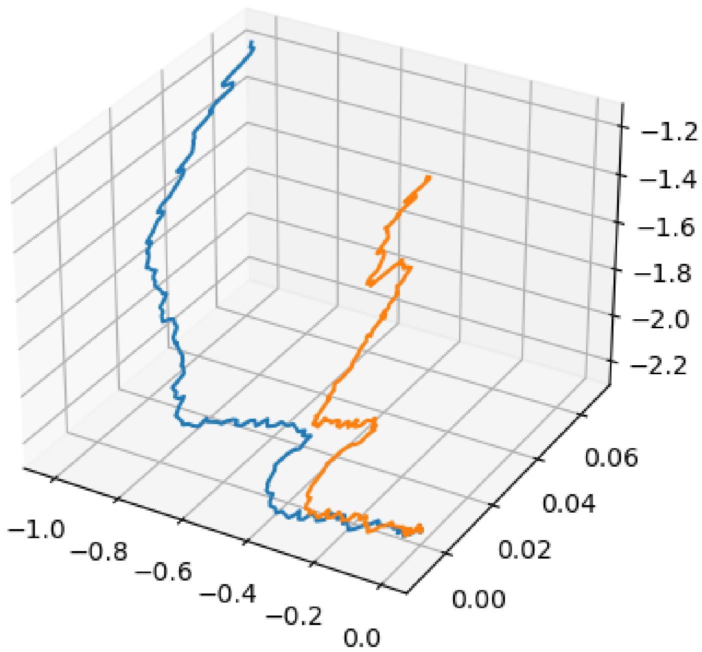

Figure 8: Ground Truth vs Estimate