16-833: Robot Localization and Mapping, Spring 2021

Homework 4-Dense SLAM with Point-based Fusion

Aaron Guan (zhongg)

1 Overview

2 Iterative Closest Point (ICP)

2.1 Projective data association

1. If a point p is projected to a vertex map, the projected point must lie within the vertex map to be a valid correspondence. Therefore, given that the u, v coordinate with a depth d are obtained after projection and the vertex map's height and width are H and W, the following conditions need to be met:

   $0 <= u < W$, $0 <= v < H$, $d > 0$

2. By filtering correspondences by a distance threshold, e.g. $|p - q| < d_{thr}$, the outliers from the correspondence can be removed to avoid corrupting the computed transformation.

2.2 Linearization

With small-angle assumption, the rotation matrix can be rewrite as:

$$\delta\mathbf{R}(\alpha,\beta,\gamma) \approx \begin{pmatrix} 1 & \alpha\beta-\gamma & \alpha\gamma+\beta & 0 \\ \gamma & \alpha\beta\gamma+1 & \beta\gamma-\alpha & 0 \\ -\beta & \alpha & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \approx \begin{pmatrix} 1 & -\gamma & \beta & 0 \\ \gamma & 1 & -\alpha & 0 \\ -\beta & \alpha & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \hat{\mathbf{R}}(\alpha,\beta,\gamma). \quad (1)$$

Then we can formulate a Matrix M:

$$\hat{\mathbf{M}} = \mathbf{T}\left(t_x, t_y, t_z\right) \cdot \hat{\mathbf{R}}(\alpha,\beta,\gamma) = \begin{pmatrix} 1 & -\gamma & \beta & t_x \\ \gamma & 1 & -\alpha & t_y \\ -\beta & \alpha & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (2)$$

Therefore, the incremental update can be written as a linear expression of the six parameters $\alpha, \beta, \gamma, t_x, t_y, t_z$:

$$\left(\hat{\mathbf{M}} \cdot \mathbf{p}_i - \mathbf{q}_i\right) \bullet \mathbf{n}_i = \left(\hat{\mathbf{M}} \cdot \begin{pmatrix} p_{ix} \\ p_{iy} \\ p_{iz} \\ 1 \end{pmatrix} - \begin{pmatrix} q_{ix} \\ q_{iy} \\ q_{iz} \\ 1 \end{pmatrix}\right) \cdot \begin{pmatrix} n_{ix} \\ n_{iy} \\ n_{iz} \\ 0 \end{pmatrix} \quad (3)$$

$$= \left[\left(n_{iz}p_{iy} - n_{iy}p_{iz}\right)\alpha + \left(n_{ix}p_{iz} - n_{iz}p_{ix}\right)\beta + \left(n_{iy}p_{ix} - n_{ix}p_{iy}\right)\gamma + n_{ix}t_x + n_{iy}t_y + n_{iz}t_z\right]$$
$$- \left[n_{ix}q_{ix} + n_{iy}q_{iy} + n_{iz}q_{iz} - n_{ix}p_{ix} - n_{iy}p_{iy} - n_{iz}p_{iz}\right]$$

Thus, given N pairs of point correspondences, we can arrange all $\left(\hat{\mathbf{M}} \cdot \mathbf{p}_i - \mathbf{q}_i\right) \bullet \mathbf{n}_i$ into a matrix expression:

$$Ax - b$$

where

$$\mathbf{b} = \begin{pmatrix} n_{1x}q_{1x} + n_{1y}q_{1y} + n_{1z}q_{1z} - n_{1x}p_{1x} - n_{1y}p_{1y} - n_{1z}p_{1z} \\ n_{2x}q_{2x} + n_{2y}q_{2y} + n_{2z}q_{2z} - n_{2x}p_{2x} - n_{2y}p_{2y} - n_{2z}p_{2z} \\ \vdots \\ n_{Nx}q_{Nx} + n_{Ny}q_{Ny} + n_{Nz}q_{Nz} - n_{Nx}p_{Nx} - n_{Ny}p_{Ny} - n_{Nz}p_{Nz} \end{pmatrix}$$

$$\mathbf{x} = \begin{pmatrix} \alpha & \beta & \gamma & t_x & t_y & t_z \end{pmatrix}^{\mathrm{T}}$$

and

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & n_{1x} & n_{1y} & n_{1z} \\ a_{21} & a_{22} & a_{23} & n_{2x} & n_{2y} & n_{2z} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{N1} & a_{N2} & a_{N3} & n_{Nx} & n_{Ny} & n_{Nz} \end{pmatrix}$$

with

$$\begin{aligned} a_{i1} &= n_{iz}p_{iy} - n_{iy}p_{iz}, \\ a_{i2} &= n_{ix}p_{iz} - n_{iz}p_{ix}, \\ a_{i3} &= n_{iy}p_{ix} - n_{ix}p_{iy}. \end{aligned}$$

2.3 Optimization

1. From the above derivation, we can know that

$$\sum_{i=1^n} r_i^2(\alpha, \beta, \gamma, t_x, t_y, t_z) = \sum_{i=1}^{n} \left\| A_i \begin{bmatrix} \alpha \\ \beta \\ \gamma \\ t_x \\ t_y \\ t_z \end{bmatrix} - b_i \right\|^2. \tag{4}$$

   and we can obtain the transformation matrix $\hat{\mathbf{M}}$ by solving for

$$\mathbf{x}_{\text{opt}} = \arg\min_{\mathbf{x}} |\mathbf{A}\mathbf{x} - \mathbf{b}|^2 \tag{5}$$

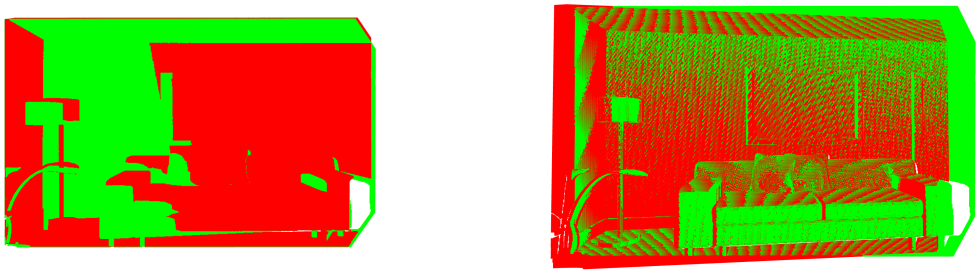
   which is a standard linear least-square problem. Taking partial derivatives with respect to the x variables and set them to 0, this gives us normal equation:

$$\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{b}$$

   We can use SVD to solve it or using QR/LU factorization to solve it. If we use Pseudo-Inverse to solve it, the solution would be:

$$\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$$

2. The visualization before and after ICP with source frame 10 and target frame 50 is shown in Figure 1.

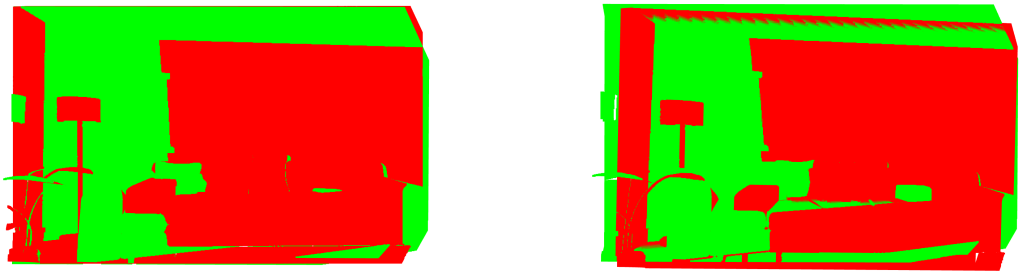


(a) Point clouds before registration

(b) Point clouds after registration

Figure 1: Point cloud before and after registration with source frame 10 and target frame 50

The visualization before and after ICP with source frame 10 and target frame 100 is shown in Figure 2.



(a) Point clouds before registration

(b) Point clouds after registration

Figure 2: Point cloud before and after registration with source frame 10 and target frame 100

From the above figures, we can know that the point cloud registration between frame 10 and frame 50 is pretty successful while the registration between frame 10 and frame 100 is not good. The reason is that the frame 10 and frame 100 has a huge difference such that the number of valid correspondences is much less and the number of noises increases. Without enough correspondences, the loss of ICP cannot be reduced further to a minimum and the transformation will not be the optimal result.

3 Point-based Fusion

3.1 Filter

The code of `filter_pass1`, `filter_pass2` are implemented in `fusion.py`.

3.2 Merge

Given $p \in R^3$ in the map coordinate system with a weight w and its corresponding point $q \in R^3$ in the frame's coordinate system with a weight 1, The weight average of the positions is:

$$\bar{p} = \frac{wp + (R_c^w q + t_c^w)}{w + 1}$$

The weight average of normals is:

$$\bar{n_p} = \frac{wn_p + R_c^w n_q}{w+1}$$

Then the unit normals are obtained by normalizing the normals after weight average.

3.3 Addition

the corresponding part in `add` is implemented in `fusion.py`.

3.4 Results

The visualization of the fusion map with a normal map is shown in Figure 3.

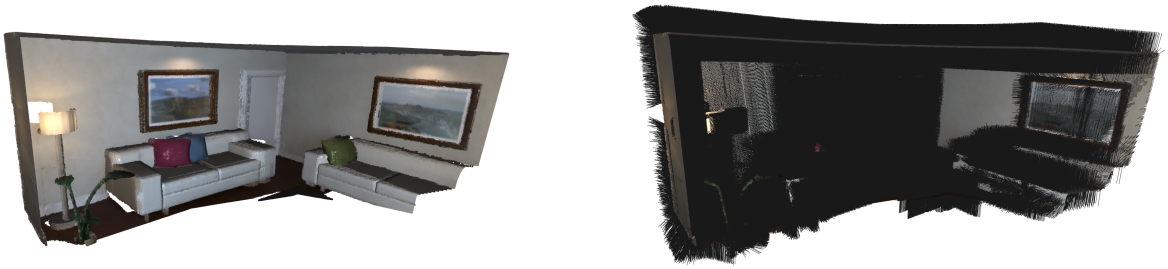


Figure 3: Fusion with ground truth poses with a normal map.

4 The dense SLAM system

1. The source is the points in the Map and the target is the current RGBD-frame at the time step. We **cannot** swap their roles because this is frame-to-model ICP and the Map points need to be projected onto the vertex map of the input RGBD-frame for correspondences matching. If we swapped the roles, we do not have the vertex map of the Map and therefore we cannot find the correspondences.

2. The fusion visualization with poses estimated from frame-to-model ICP is shown in Figure 4 and the estimated trajectory with the ground truth is shown in Figure 5.

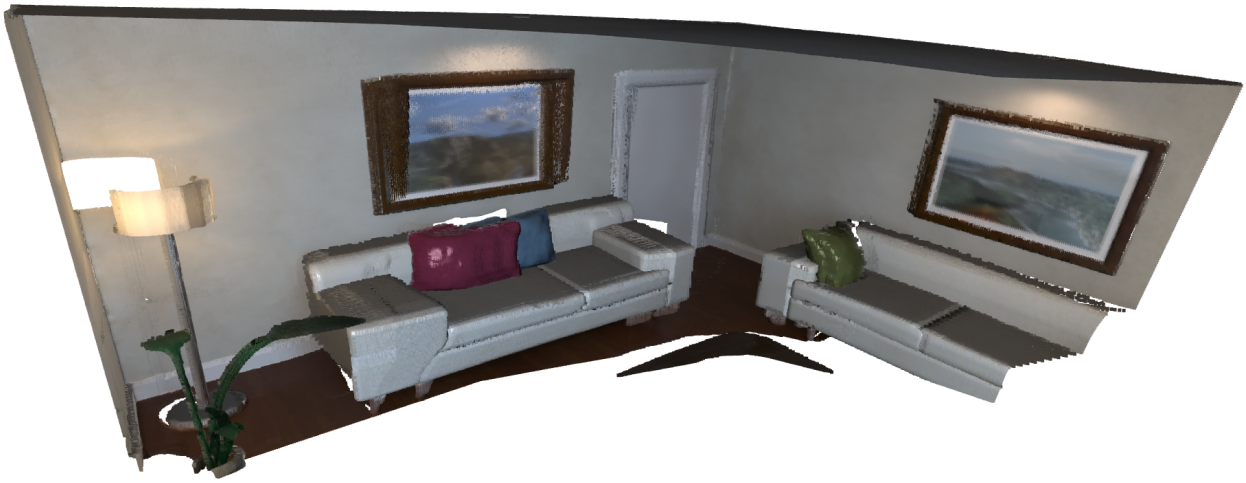


Figure 4: Fusion with poses estimated from frame-to-model ICP.

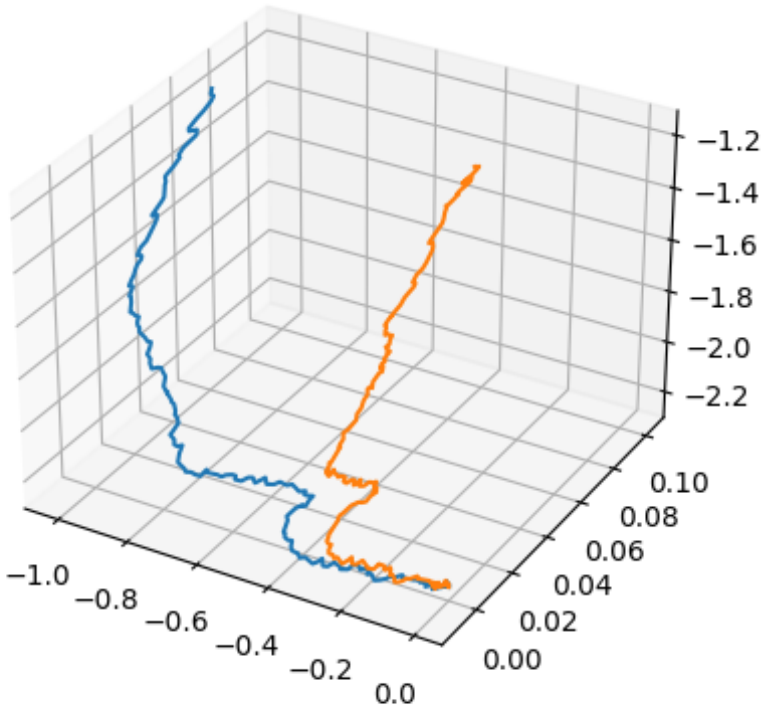


Figure 5: Estimated trajectory with ground truth

From the fusion visualization we can observe there is small drift in the fusion map. As shown in the Figure 5, the estimated trajectory also has a drift with the ground truth trajectory while they have very similar shape. Therefore, it has a good estimation in local transformation but the error can be accumulated, which causes drift in the global trajectory.

3. **Bonus**

   To reduce the drift, I removed the points that remain in the unstable state for a long time. These dormant points are likely outliers or artifacts from moving objects and will be removed after a certain period. Therefore, a weight threshold c_{stable} is specified such that points with weight $c_k < c_{stable}$ are considered as unstable. Meanwhile, the times of points added in the map are also recorded. After a certain period T_{max}, those unstable points will be removed from the map.

   The implementation details is in `remove_outliers` in `fusion.py`. Note that by default, the outliers removal is enable for the fusion. The fusion result after outliers removal is shown in Figure 6. Comparing with Figure 4, we can clearly observe the drift is reduced. However, some structure details of object are removed. To improve it, a better weight threshold and other methods for outlier removal might need to be considered in the future.

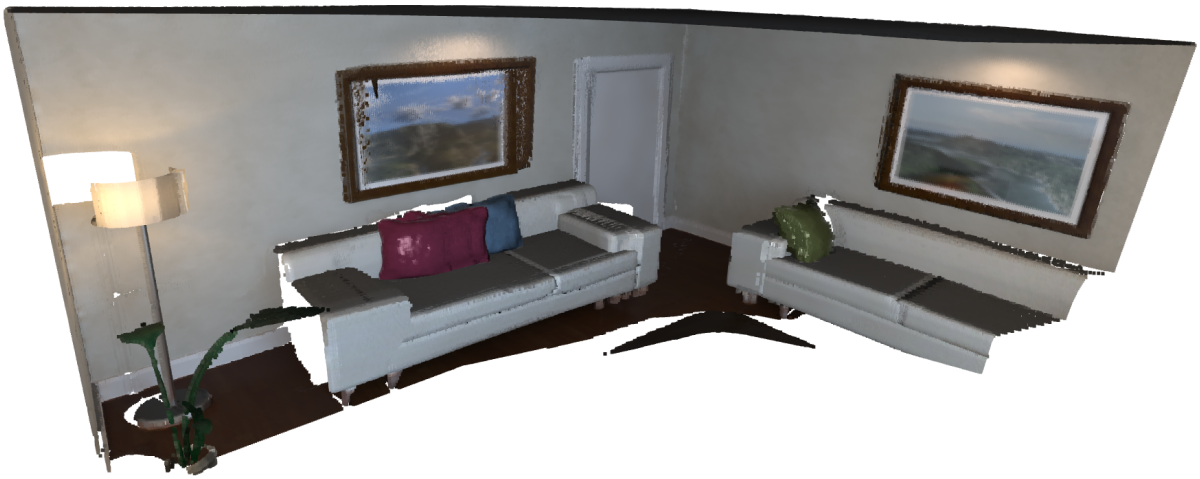


Figure 6: Fusion with outliers removal.