[16-833] Homework 4 : Written Report

Bharath Somayajula

April 14, 2023

Contents

1 Iterative Closest Point

1.1 Projective Data Association

1.1.1 Conditions

For a source point to have a valid correspondence in target vertex map, it's projected coordinates must fall within the bounds of the vertex map and have a positive depth. Therefore, assuming that the u, v coordinates are rounded to integers and are zero-indexed, we get:

$$0 \leq u < W$$

$$0 \leq v < H$$

$$d > 0$$

1.1.2 Distance Filtering

Distance filtering is needed for a few reasons:

1. We assume that change in pose is small between consecutive measurements. This implies that change in vertex map between consecutive frames is also small. So it makes sense to apply distance thresholds
2. Vertex map is constructed from depth map and depth measurements can be corrupted for several reasons such as noise, wildly inaccurate measurements due to surface properties of materials whose depth is being measured or in practical settings due to objects entering the scene between frames. Therefore, applying distance threshold is a reasonable way to overcome these issues.

1.2 Linearization

The expression $(\delta R)p_i^{'} + \delta t$ needs to be simplified first.

$$
\begin{aligned}
(\delta R)p_i^{'} + \delta t &= \begin{bmatrix} 1 & -\gamma & \beta \\ \gamma & 1 & -\alpha \\ -\beta & \alpha & 1 \end{bmatrix} \times \begin{bmatrix} p_{ix}^{'} \\ p_{iy}^{'} \\ p_{iz}^{'} \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \\
&= \begin{bmatrix} p_{ix}^{'} - \gamma p_{iy}^{'} + \beta p_{iz}^{'} + t_x \\ \gamma p_{ix}^{'} + p_{iy}^{'} - \alpha p_{iz}^{'} + t_y \\ -\beta p_{ix}^{'} + \alpha p_{iy}^{'} + p_{iz}^{'} + t_z \end{bmatrix} \\
&= \begin{bmatrix} 0 & p_{iz}^{'} & -p_{iy}^{'} & 1 & 0 & 0 \\ -p_{iz}^{'} & 0 & p_{ix}^{'} & 0 & 1 & 0 \\ p_{iy}^{'} & -p_{ix}^{'} & 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \alpha \\ \beta \\ \gamma \\ t_x \\ t_y \\ t_z \end{bmatrix} + \begin{bmatrix} p_{ix}^{'} \\ p_{iy}^{'} \\ p_{iz}^{'} \end{bmatrix} \\
&= \left[[-p_i^{'}]_\times | I_{3\times 3} \right] x + p_i^{'}
\end{aligned}
$$

Substituting in the objective function, we get:

$$
\begin{aligned}
&= \sum_i ||n_{q_i}^T \left[[-p_i^{'}]_\times | I_{3\times 3} \right] x + n_{q_i}^T p_i^{'} - n_{q_i}^T q_i||^2 \\
&= \sum_i ||A_i x + b_i||^2
\end{aligned}
$$

Therefore

$$
A_i = n_{q_i}^T \left[[-p_i^{'}]_\times | I_{3\times 3} \right] \text{ and } b_i = n_{q_i}^T (p_i^{'} - q_i)
$$

1.3 Optimization

1.3.1 Linear System

The A_i and b_i can be concatenated along row dimension to obtain matrices A and b where

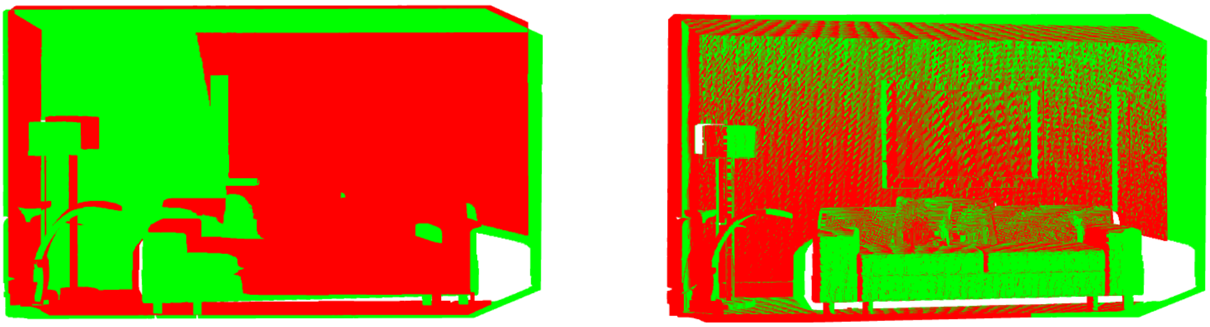
$$
A = \begin{bmatrix} A_1 \\ A_2 \\ ... \\ A_n \end{bmatrix}_{n\times 6} \quad \text{and } b = \begin{bmatrix} b_1 \\ b_2 \\ ... \\ b_n \end{bmatrix}_{n\times 1}
$$

and the objective function is

$$||Ax + b||^2$$

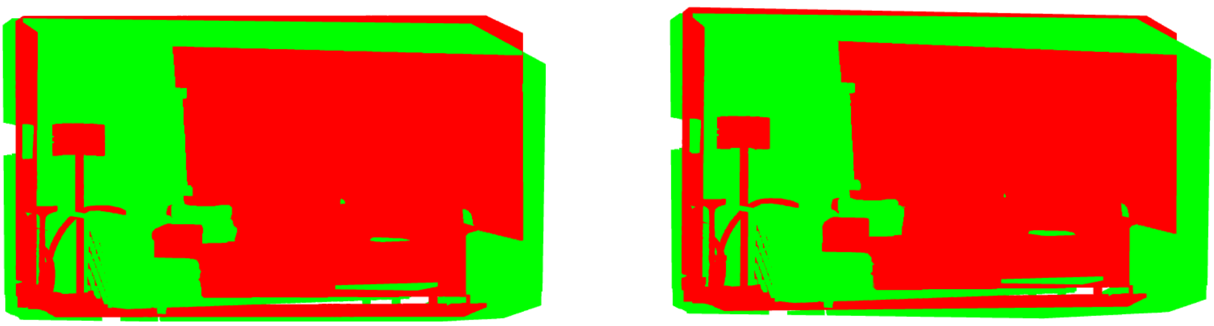
which can be solved using least squares solution for $x = (A^T A)^{-1} A^T (-b)$ (**NOTE:** $-b$ since the problem is formulated in handout as $||Ax + b||^2$ instead of $||Ax - b||^2$).

1.3.2 Results



(a) Before (b) After

Figure 1: Results with frames 10 and 50



(a) Before (b) After

Figure 2: Results with frames 10 and 100

The pose estimation fails for frames 10 and 100 because the small pose change assumption is violated.

2 Point-based Fusion

2.1 Merge

The merge operation for points and normals can be written as

$$p \leftarrow \frac{w \times p + (R_c^w \times q + t_c^w)}{w + 1}$$

$$n_p \leftarrow \frac{w \times n_p + R_c^w \times n_q}{w + 1}$$

$$n_p \leftarrow \frac{n_p}{||n_p||_2}$$

2.2 Results

Figure 3: RGB result

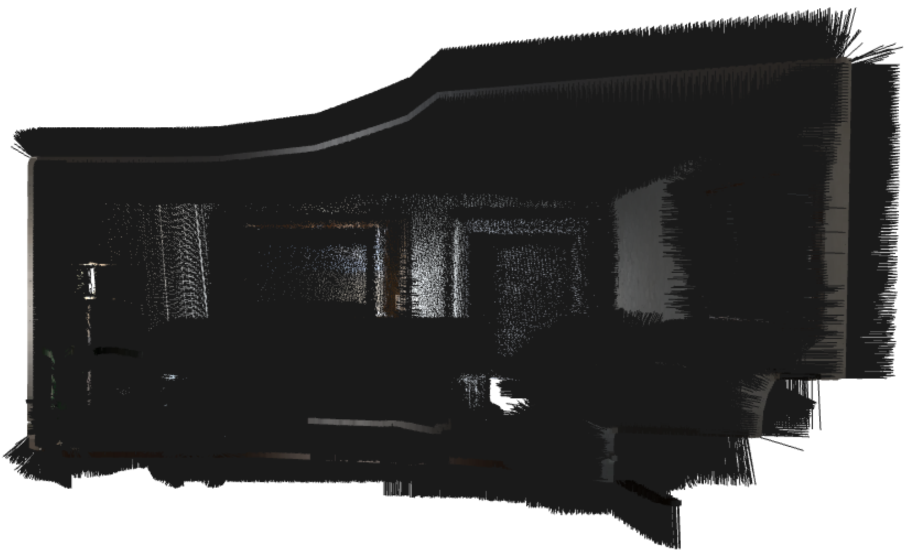


Figure 4: Normals result

Number of points is after fusion is 1362157. The number of points obtained by simple concatenation is 15360000. Therefore, the compression ratio is:

$$\text{Compression Ratio} = \frac{\text{No. of points before fusion}}{\text{No. of points after fusion}} = \frac{15360000}{1362157} = 11.28$$

3 The dense SLAM system

3.1 Source and Target

The map is the source and the RGBD input is the target.

One reason why swapping their roles will not work is that in projective data association step of ICP, we first project source points into image plane of target to associate every source point to a target point. Filtering is then performed to remove associations where (u, v) coordinates go out of bounds of target image. These steps are made possible by the fact that target is a regular 2D array of size $H \times W$ and is formed by a camera with known intrinsic parameters. The association and filtering are much harder to perform when target is an unordered point cloud which is the case with the map we store.

3.2 Results

3.2.1 RGB

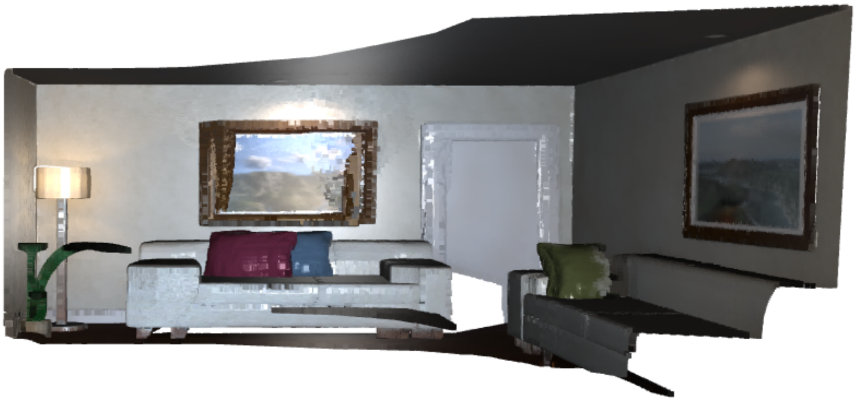


Figure 5: RGB result

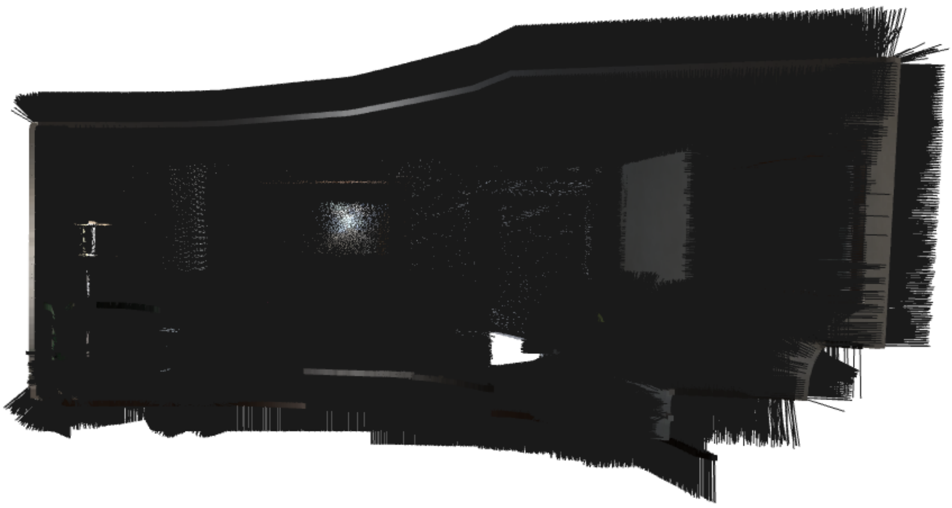


Figure 6: Normals result

3.2.2 Pose

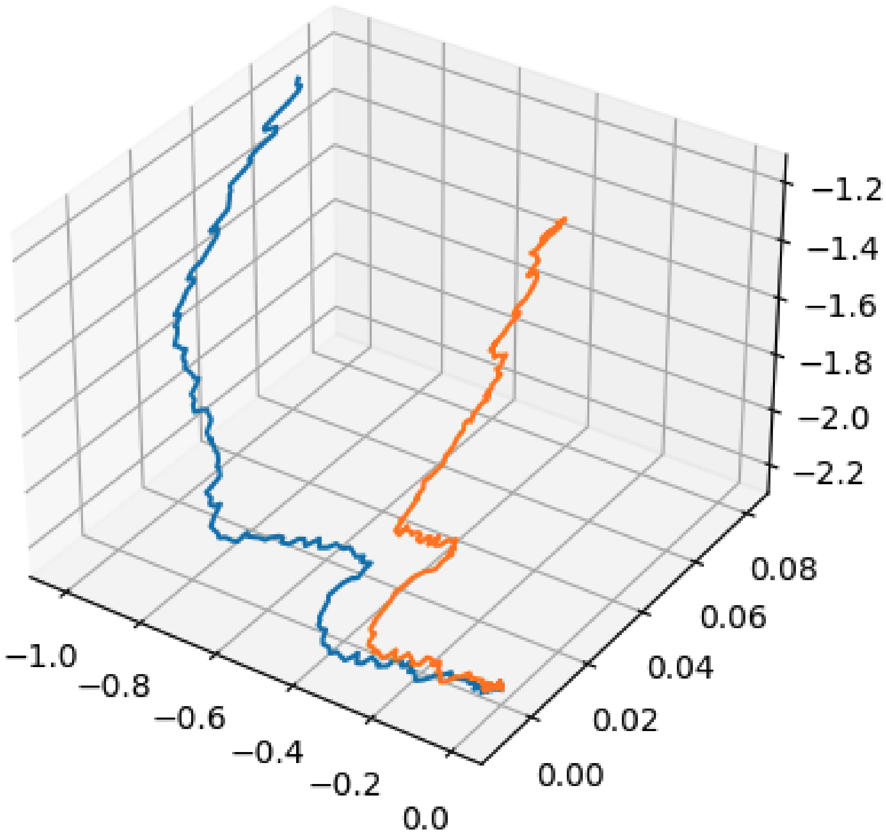


Figure 7: Drift

3.3 [BONUS] Reducing Drift

The drift has been reduced by simply using the top 10% most stable points (measured using *weights* variable of points in map) for measuring pose using ICP. The change has been implemented in *main.py* as shown below and can be uncommented to verify the implementation.

```
if len(m.points) > 2000:
        #sort by weights
        wt_sort_idx = np.argsort(m
.weights.flatten())


#select top 10% weight points
        num_points = int(0.1*len(m
.points))

        indices = wt_sort_idx[-
num_points:]
```

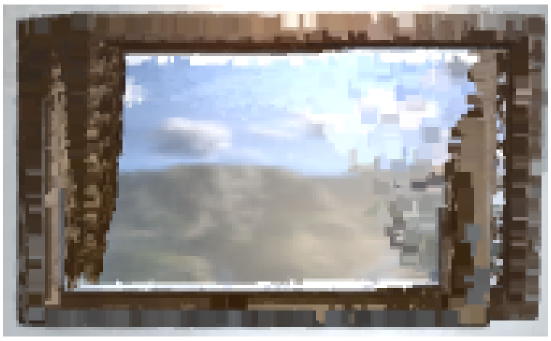
Figure 8: Code to reduce drift

The comparison of RGB, normals and poses computed with and without this change are shown below. The visualizations clearly show that this simple change not only improved the quality of results but also reduced inference time since fewer points are being used in ICP.

(a) Before change

(b) After change

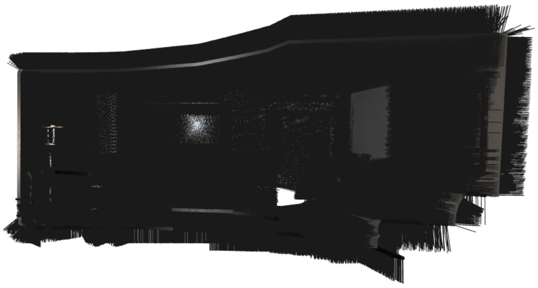
Figure 9: Comparison of RGB renderings before and after reducing drift



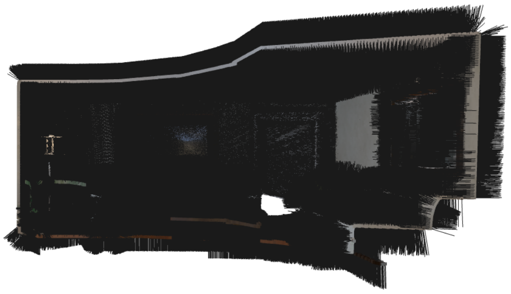
(a) Before change

(b) After change

Figure 10: Comparison of zoomed in RGB renderings before and after reducing drift

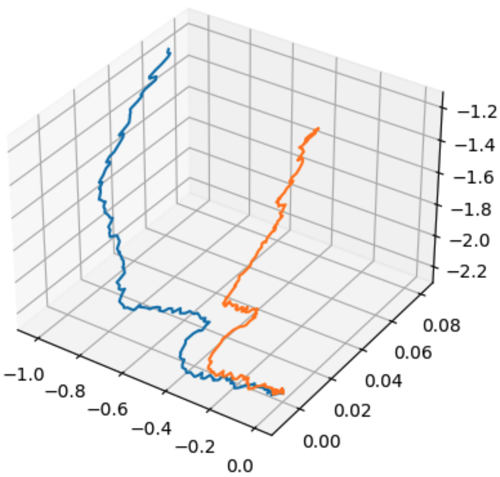


(a) Before change

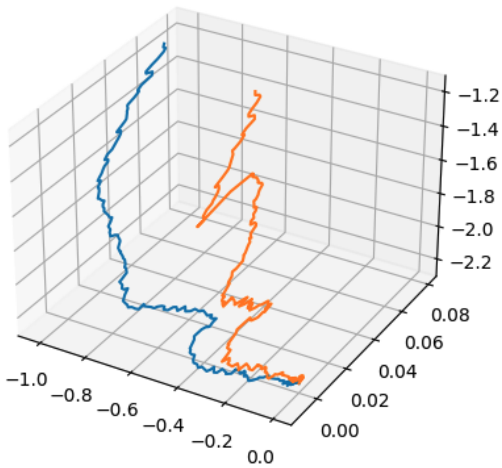


(b) After change

Figure 11: Comparison of normals before and after reducing drift



(a) Before change



(b) After change

Figure 12: Comparison of poses before and after reducing drift

The losses (at the 9^{th} iteration) for the dense SLAM before and after the change are plotted on the same graph below. With the exception of a few frames, the loss for the new method is lower than the loss for default implementation.

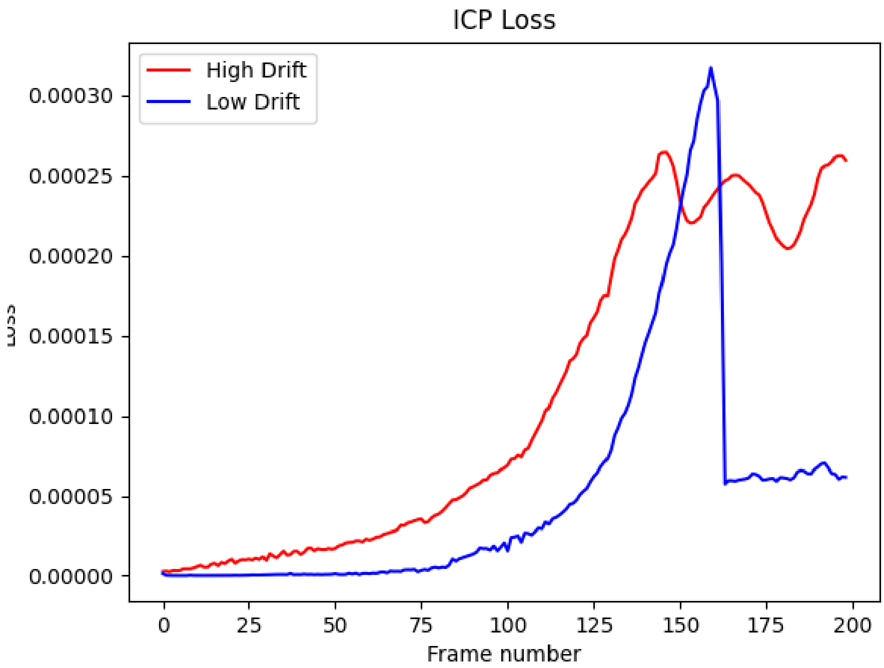


Figure 13: Comparison of losses