Adam Driscoll 4/22/2018
SLAM HW4

Collaborators: David Evans (dje1@)

ICP

A. Point-to-point ICP finds corresponding points in two frames and tries to minimize the distance between them. Point-to-plane finds corresponding points in two frames but draws a plane tangent to the point in one of the frames and tries to minimize the distance between the point and that plane.

B. To derive equation 20 from the paper we must perform the following steps:

$$\tilde{R}^z \tilde{\boldsymbol{V}}_{\boldsymbol{k}}^{\boldsymbol{g}}(\boldsymbol{u}) + \tilde{\boldsymbol{t}}^{\boldsymbol{z}} = \boldsymbol{G}(\boldsymbol{u})\boldsymbol{x} + \tilde{\boldsymbol{V}}_{\boldsymbol{k}}^{\boldsymbol{g}}(\boldsymbol{u})$$

From the paper we are given that:

$$\tilde{R}^z = \begin{bmatrix} 1 & \alpha & -\gamma \\ -\alpha & 1 & \beta \\ \gamma & -\beta & 1 \end{bmatrix}, \quad \tilde{\boldsymbol{t}}^{\boldsymbol{z}} = \begin{bmatrix} \boldsymbol{t_x} \\ \boldsymbol{t_y} \\ \boldsymbol{t_z} \end{bmatrix}$$

And we know that $\tilde{\boldsymbol{V}}_{\boldsymbol{k}}^{\boldsymbol{g}}(\boldsymbol{u})$ is a vector of points which we can represent as such:

$$\tilde{\boldsymbol{V}}_{\boldsymbol{k}}^{\boldsymbol{g}}(\boldsymbol{u}) = \begin{bmatrix} \tilde{\boldsymbol{V}}_{\boldsymbol{k,x}}^{\boldsymbol{g}} \\ \tilde{\boldsymbol{V}}_{\boldsymbol{k,y}}^{\boldsymbol{g}} \\ \tilde{\boldsymbol{V}}_{\boldsymbol{k,z}}^{\boldsymbol{g}} \end{bmatrix}$$

Multiplying through we get:

$$\begin{bmatrix} \tilde{\boldsymbol{V}}_{\boldsymbol{k,x}}^{\boldsymbol{g}} + \alpha\tilde{\boldsymbol{V}}_{\boldsymbol{k,y}}^{\boldsymbol{g}} - \gamma\tilde{\boldsymbol{V}}_{\boldsymbol{k,z}}^{\boldsymbol{g}} + \boldsymbol{t_x} \\ -\alpha\tilde{\boldsymbol{V}}_{\boldsymbol{k,x}}^{\boldsymbol{g}} + \tilde{\boldsymbol{V}}_{\boldsymbol{k,y}}^{\boldsymbol{g}} + \beta\tilde{\boldsymbol{V}}_{\boldsymbol{k,z}}^{\boldsymbol{g}} + \boldsymbol{t_y} \\ \gamma\tilde{\boldsymbol{V}}_{\boldsymbol{k,x}}^{\boldsymbol{g}} - \beta\tilde{\boldsymbol{V}}_{\boldsymbol{k,y}}^{\boldsymbol{g}} + \tilde{\boldsymbol{V}}_{\boldsymbol{k,z}}^{\boldsymbol{g}} + \boldsymbol{t_z} \end{bmatrix}$$
$$= \begin{bmatrix} \alpha\tilde{\boldsymbol{V}}_{\boldsymbol{k,y}}^{\boldsymbol{g}} - \gamma\tilde{\boldsymbol{V}}_{\boldsymbol{k,z}}^{\boldsymbol{g}} + \boldsymbol{t_x} \\ -\alpha\tilde{\boldsymbol{V}}_{\boldsymbol{k,x}}^{\boldsymbol{g}} + \beta\tilde{\boldsymbol{V}}_{\boldsymbol{k,z}}^{\boldsymbol{g}} + \boldsymbol{t_y} \\ \gamma\tilde{\boldsymbol{V}}_{\boldsymbol{k,x}}^{\boldsymbol{g}} - \beta\tilde{\boldsymbol{V}}_{\boldsymbol{k,y}}^{\boldsymbol{g}} + \boldsymbol{t_z} \end{bmatrix} + \begin{bmatrix} \tilde{\boldsymbol{V}}_{\boldsymbol{k,x}}^{\boldsymbol{g}} \\ \tilde{\boldsymbol{V}}_{\boldsymbol{k,y}}^{\boldsymbol{g}} \\ \tilde{\boldsymbol{V}}_{\boldsymbol{k,z}}^{\boldsymbol{g}} \end{bmatrix}$$

Again, from the paper we are given that:

$$\boldsymbol{G}(\boldsymbol{u}) = \left[\left[\tilde{\boldsymbol{V}}_{\boldsymbol{k}}^{\boldsymbol{g}}(\boldsymbol{u}) \right]_\times | \, \boldsymbol{I_{3x3}} \right]$$
$$= \begin{bmatrix} 0 & -\tilde{\boldsymbol{V}}_{\boldsymbol{k,z}}^{\boldsymbol{g}} & \tilde{\boldsymbol{V}}_{\boldsymbol{k,y}}^{\boldsymbol{g}} & \boldsymbol{1} & \boldsymbol{0} & \boldsymbol{0} \\ \tilde{\boldsymbol{V}}_{\boldsymbol{k,z}}^{\boldsymbol{g}} & \boldsymbol{0} & -\tilde{\boldsymbol{V}}_{\boldsymbol{k,x}}^{\boldsymbol{g}} & \boldsymbol{0} & \boldsymbol{1} & \boldsymbol{0} \\ -\tilde{\boldsymbol{V}}_{\boldsymbol{k,y}}^{\boldsymbol{g}} & \tilde{\boldsymbol{V}}_{\boldsymbol{k,x}}^{\boldsymbol{g}} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{1} \end{bmatrix}$$

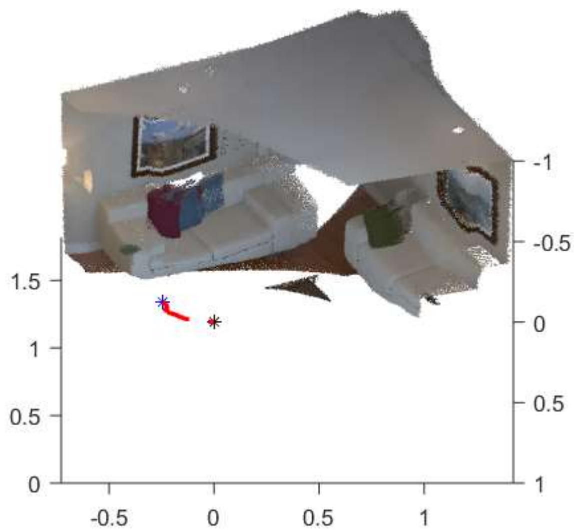
$$\boldsymbol{x} = \begin{bmatrix} \beta \\ \gamma \\ \alpha \\ t_x \\ t_y \\ t_z \end{bmatrix}$$

And multiplying these through and adding the points we get:

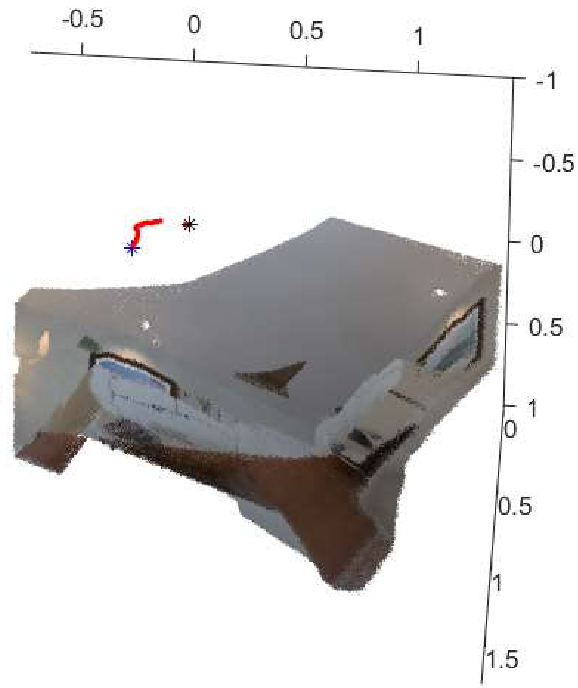
$$\begin{bmatrix} \alpha\tilde{\boldsymbol{V}}_{\boldsymbol{k,y}}^{\boldsymbol{g}} - \gamma\tilde{\boldsymbol{V}}_{\boldsymbol{k,z}}^{\boldsymbol{g}} + \boldsymbol{t_x} \\ -\alpha\tilde{\boldsymbol{V}}_{\boldsymbol{k,x}}^{\boldsymbol{g}} + \beta\tilde{\boldsymbol{V}}_{\boldsymbol{k,z}}^{\boldsymbol{g}} + \boldsymbol{t_y} \\ \gamma\tilde{\boldsymbol{V}}_{\boldsymbol{k,x}}^{\boldsymbol{g}} - \beta\tilde{\boldsymbol{V}}_{\boldsymbol{k,y}}^{\boldsymbol{g}} + \boldsymbol{t_z} \end{bmatrix} + \begin{bmatrix} \tilde{\boldsymbol{V}}_{\boldsymbol{k,x}}^{\boldsymbol{g}} \\ \tilde{\boldsymbol{V}}_{\boldsymbol{k,y}}^{\boldsymbol{g}} \\ \tilde{\boldsymbol{V}}_{\boldsymbol{k,z}}^{\boldsymbol{g}} \end{bmatrix}$$

Which is of the same form as the equation above.

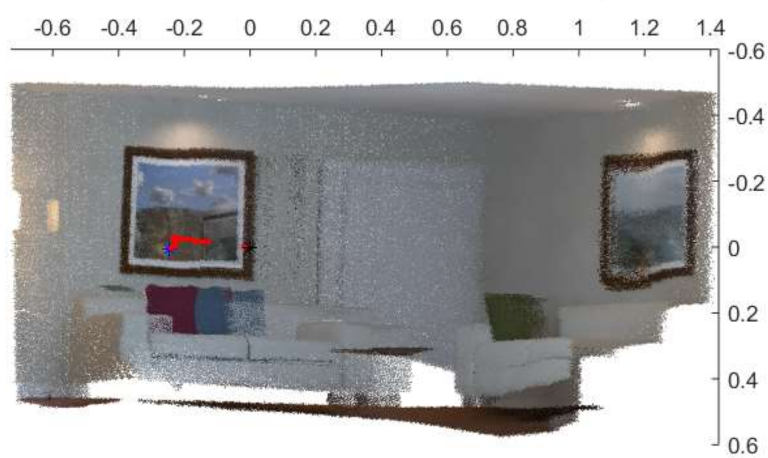
D. Results:



A.



B.



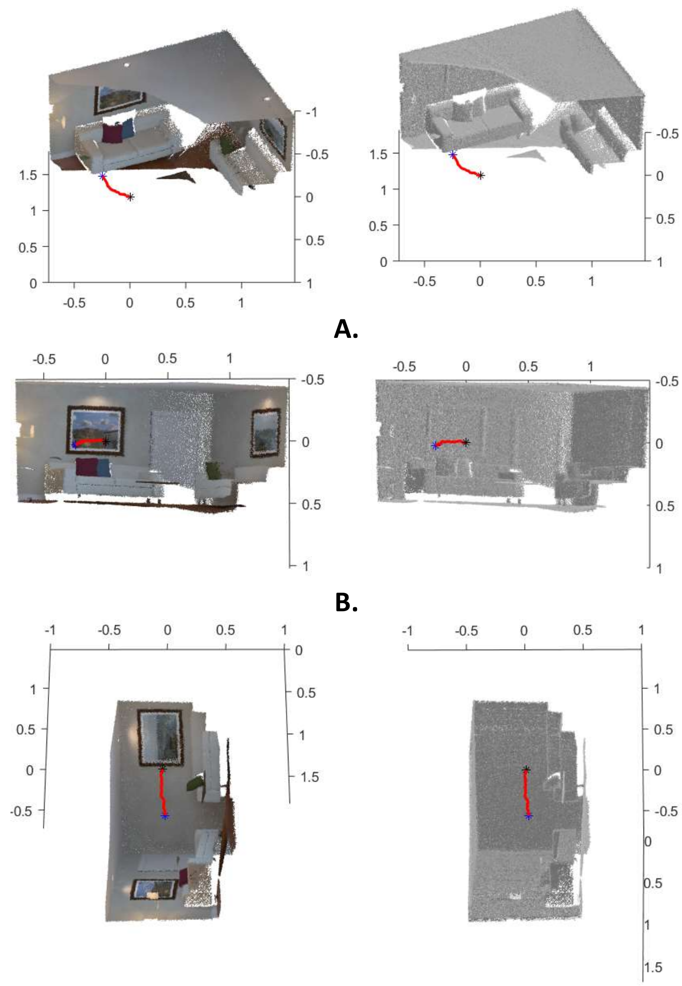
C.

Figure 1 A-C: Results of the ICP algorithm with current frame fusion

In Figure 1 above we can see the reconstruction of a small room using the Iterative Closest Point algorithm. The red line represents the estimated camera trajectory, with the black asterisk representing the starting position and the blue the final position. A discontinuity in the camera trajectory can be noticed. Within the reconstruction there are some inaccuracies present, the blue pillow on the couch was not fully reconstructed and looks like a second version of it was partially built. This is likely related to the gap in the camera trajectory, we lost track of where the camera was for some period of time, so the reconstruction shifted. The paintings on the wall are also wavy instead of straight. The reconstruction is also grainy in some areas.

Point-based Fusion

A. In general, point-based fusion methods reduce the computational complexity and memory overhead of volumetric fusion techniques.
B. $\boldsymbol{n}' = R\boldsymbol{n}$
D. Results:



A. B. C.

Figure 2 A-C: Results of the ICP algorithm with point-based fusion

In Figure 2 above we can see the 3D reconstruction of a room using the ICP algorithm, this time using point-based fusion. Comparing these results to those seen in Figure 1, many of the inaccuracies have been eliminated, though some error remains. The estimated camera trajectory is now continuous, the gap from figure has been closed. As a result, the “doubling” of certain aspects of the construction (i.e. the blue pillow) has been eliminated and the picture borders are now uniform. There are still gaps and general sparsity in the reconstruction, which is likely due to just not having the necessary data given that this is a short video clip.

Results and Discussion

A. Using the fusion model as the next reference frame results in noticeably better registration than just using the current frame as the next reference. This is because the fusion model is constantly being updated with the best points to use for the reconstruction and uses those rather than just using what we have right now which may be inaccurate.

B. We can validate the performance of the algorithm by verifying that inliers remains relatively high RMSE becomes very low throughout the execution of the program. Inliers remaining high means that we're successfully tracking points between frames as we are processing new frames. RMSE being low means we are doing a successful job of deriving the transformation between each frame and thus the correspondences between each frame are being calculated at roughly the same position. The compression ratio can be used to determine how many points the algorithm has decided to keep and track. A low compression ratio points to the algorithm removing redundant points and thus improving memory efficiency. However, a ratio that's too low means that the algorithm has started to remove important points and the resulting reconstruction will become sparse. A ratio that's too high means that the algorithm has attributed high importance to a lot of points which will result in point redundancy in the final point cloud.

C. Eval Results:

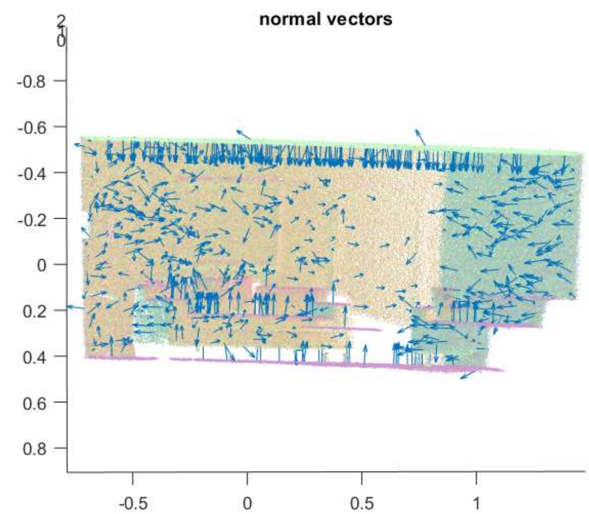


Figure 3: Normal Vectors of Reconstruction

Figure 3 shows the estimated normal vectors to various points within the point cloud. Each planar surface in the construction is represented with a different color. The more uniform planes look to have more accurate normal predictions. The wall on the left side looks to have some error in the normal prediction, possibly getting thrown off by the image that is on the wall.

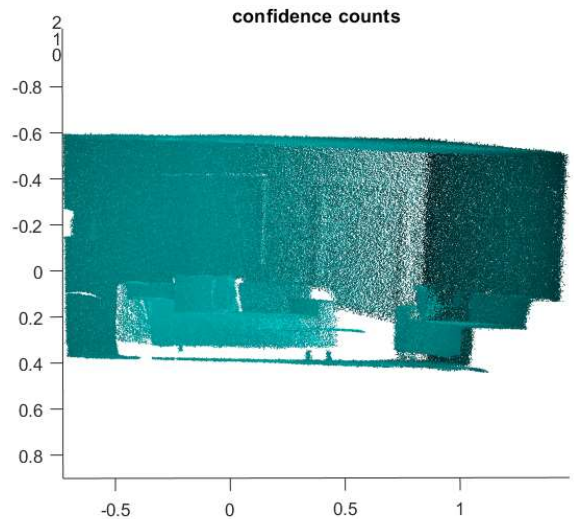


Figure 4: Confidence Counts Visualization

Figure 4 shows a visualization of the confidence counts, lighter being a higher value, darker being lower. This makes sense as if you watch the actual video file, the camera is positioned on the left side of the room making the left wall closer than the right wall. The confidence of each point is based off that points distance from the camera, further away being lower confidence.

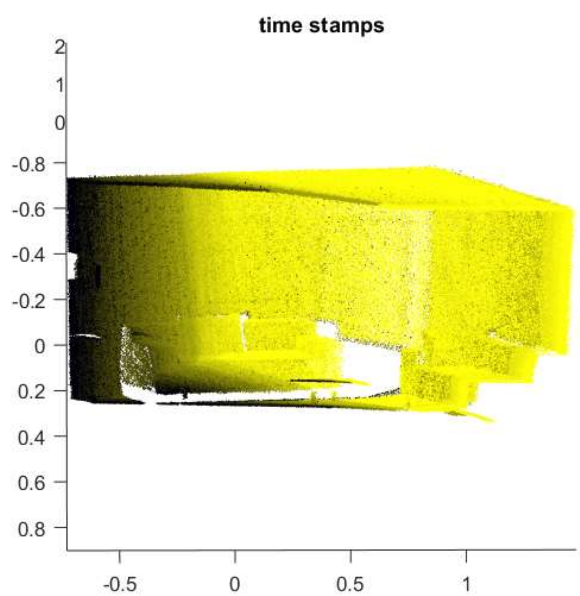


Figure 5: Timestamp visualization

Figure 5 shows a visualization of the timestamp of each point, again lighter being higher and darker being lower. Watching the video, the camera starts pointing at the left side of the room, so these points are added first, at a low timestamp. Then the camera pans right adding the points on the right side of the room at a higher time stamp.