Advanced Vision Practical Report

S1778365 S1732091

Abstract

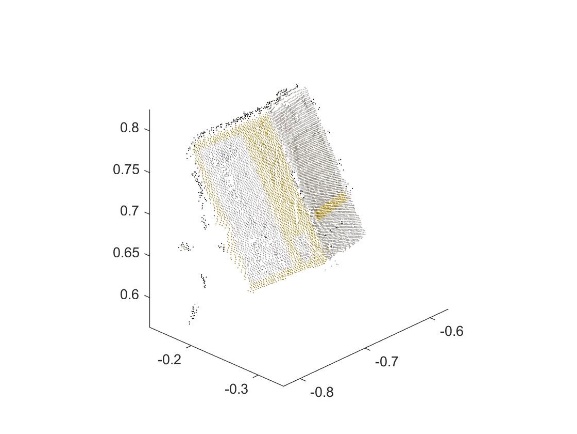
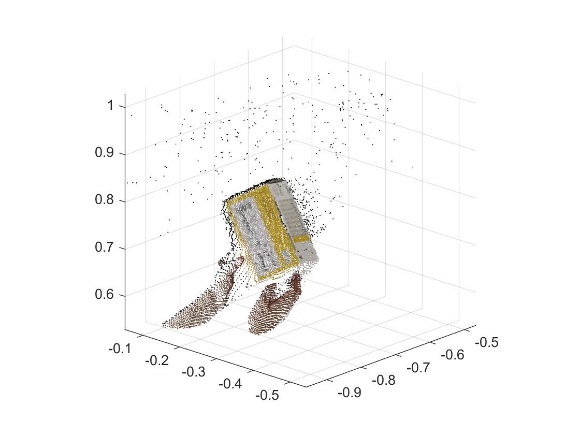
Introduction

3D model construction has been widely used in many applications like games, films, and toys manipulation. In this report, we are going to construct a 3d box using range data obtained by Kinect sensor. There are 50 frames in the dataset, and we chose 20 frames that contain valuable information including point clouds of at lease two planes. The other 30 frames contains information about only one plane or blurred positions. They are frame 6, 9, 12, 15, 18, 21, 22, 24, 27, 28, 30, 31, 33, 36, 39, 40, 43, 46, 47. Then we implemented noise removal methods to acquire the information about the box. After we extracted the box in each frame, we extracted the planes in relevant data in each frame. Using the information about planes, we also extracted the position of corners, so that we can estimate the rotation and translation between two consecutive point clouds. Normally, we use 2 surface normal and one line normal to calculate the rotation matrix, and use the mean of plane in consecutive frames to calculate the translation matrix. After obtaining all the rotation and translation between 20 frames, then we can fuse the data all together and evaluate the acquired model.

Methodology

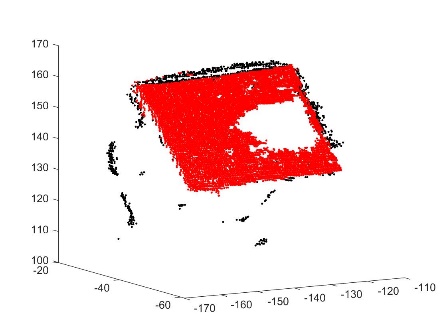
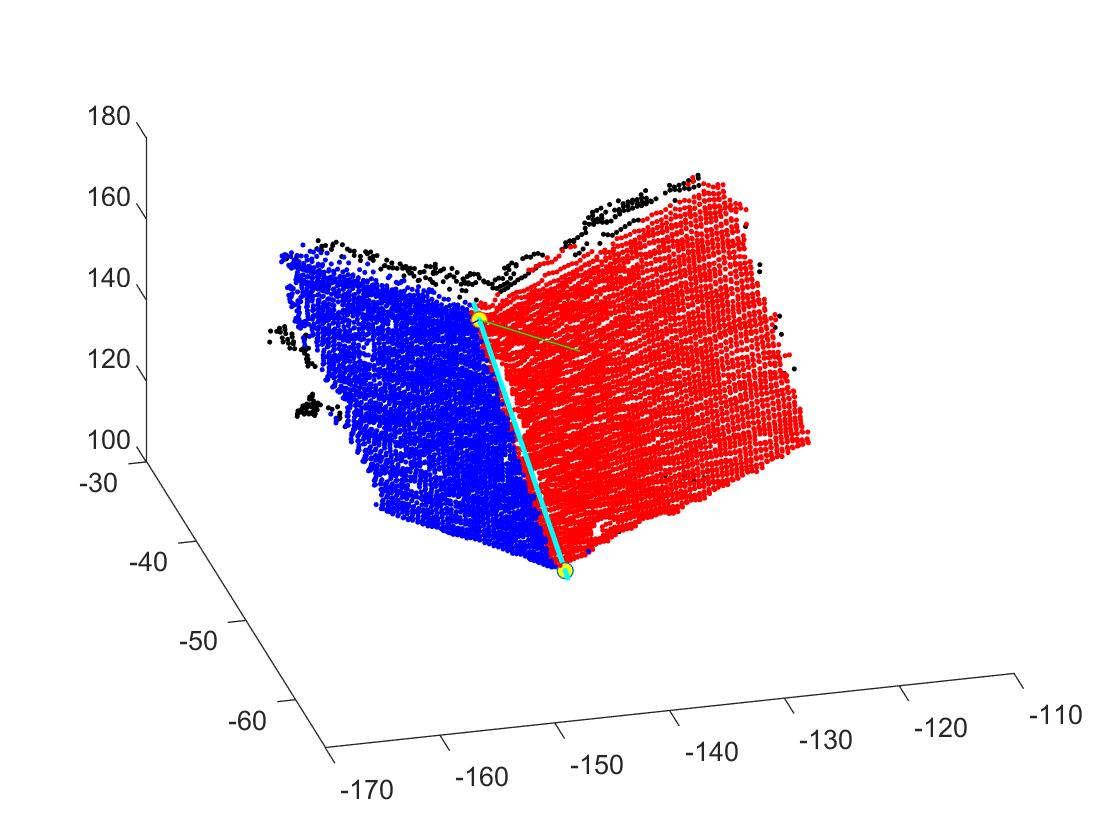
1. Noise removal

We remove the noise in three steps. First, we remove all the points without values (save computation time). And then we remove those points that are far from the box as the box is almost locate at (-0.71, -0.30,0.81) all the time. We crop the range data with x,y,z distance limit of 0.25. And the result is shown in Fig. $. It is clear shown that there are still many range data points of hands and background noise as well. To clear up the data point created by hand, we simply apply RGB threshold to the points cloud. Actually, we have tried using other methods like HSV and normalised RGB, but results in worse output. At last, considering background noise data points are scattered and sparse, we apply k-neighbor algorithm to remove them. The basic idea is to compute the distance among all the data points, and for each data point, if it does not have N neighbor data points that are withinτdist , then we will just remove it.



1. Plane extraction

The methodology we used in this section is exactly what we have learned in the lectures--The Planar Segmentation Algorithm. To explain the algorithm in our own words, we would like to separate it into two stages. The first stage is to randomly select a small patch of data points that are close to each other and can be well-fitted in one single plane. There are 3 key hyperparameters that will influence the performance of the algorithm, which are the thresholds of maximum data points’ distance, least number of data points in the patch and the plane fitting error. If the maximum distance between selected data points and least data point number is increased, the plane will be more sensitive to the noise. The second stage is to include more data points which are close to the plane ( |n’y + d|<τp ) we generate at stage 1 and are close to at least one data point from the original patch (|| y-z||<τn). This stage will keep going until the new fitted plane has a large error that exceeds the threshold.



1. Corner extraction

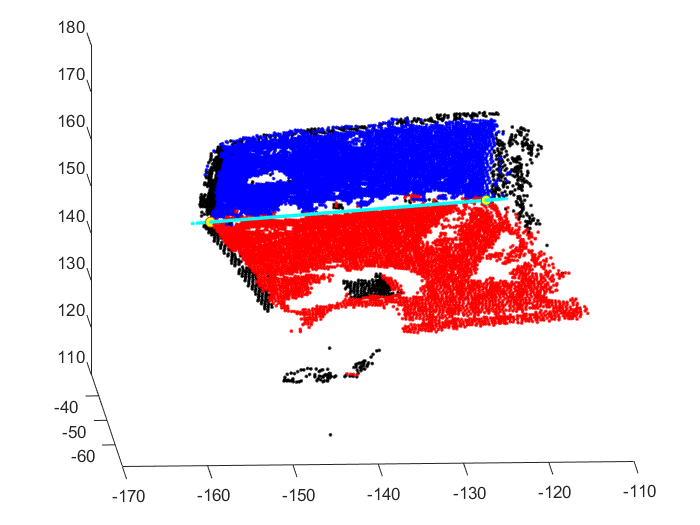
As most of the frames can see only one to two planes of the box, we mainly use two planes to extract the corners. As illustrated in Fig$, the two planes extracted have a clear intersection line. We can extract the corners by adapting two projections. The first projection is to project the data points onto the fitted plane by the formula:

q\_proj = q - dot(q - p, n) \* n

where q is the point we want to project, n is the normal vector of the plane and p is a point on the plane. And then we project all the projected points on these two planes onto the intersection line by the formula:

q\_proj = dot(q-p, n)\*n+p;

Then we can draw the intersection line between two planes as well as two corners.

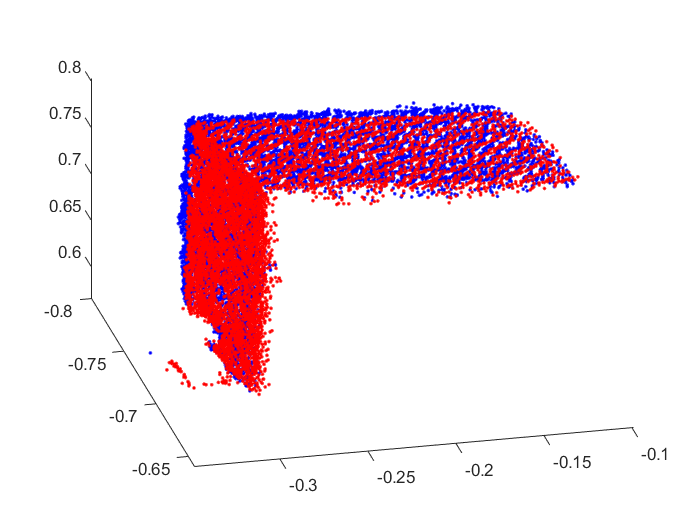


1. Rotation and translation estimation

For rotation, we use provided from the course webpage which is obtained by computing the singular value decomposition. But for translation, we use the coordinates of one corner to calculate the translation for those consecutive plane:

T = -R\*Corner\_frameA +Corner\_frameB

In the case of only one common plane is between two frames, we calculate the mean of that common plane in both frame and use them to calculate the translation matrix.



1. Fuse of 3D data

As many frames contain only one plane and we did not find out an efficient way to recognize the different faces of the box, we just fuse the point cloud manually. The results are shown below.

1. Model quality evaluation

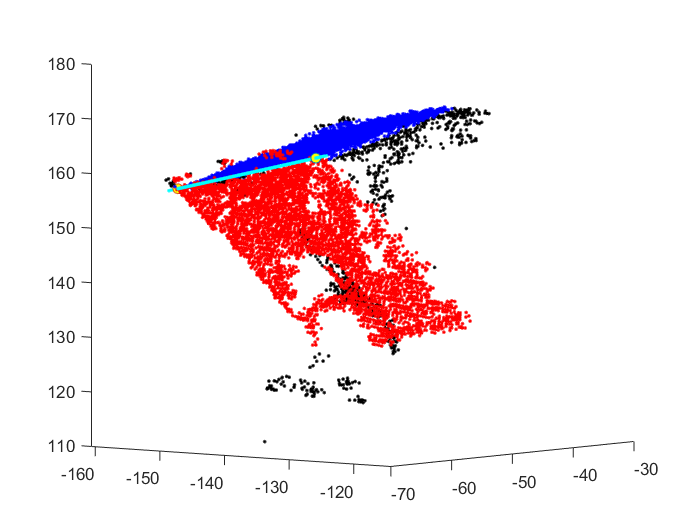
Experiment Results

1. Extract the relevant data from each point cloud
2. Extract the planes from each point cloud

The original point could is scaled by 200 in our experiments in for convenience of adjusting hyperparameters.

|  |  |  |  |
| --- | --- | --- | --- |
| **Image name** | **Number of 3D box pixels** | **Number of planes** | **Average plane fit RMS** |
| 1 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 |
| 3 | 1115 | 1 | 0.481558 |
| 4 | 6755 | 1 | 0.424857 |
| 5 | 7980 | 1 | 0.390564 |
| 6 | 10650 | 2 | 0.312606 |
| 7 | 2462 | 0 | 0 |
| 8 | 2366 | 0 | 0 |
| 9 | 11109 | 2 | 0.306806 |
| 10 | 9576 | 0 | 0 |
| 11 | 9285 | 2 | 0.296059 |
| 12 | 7681 | 3 | 0.353043 |
| 13 | 3545 | 3 | 0.418837 |
| 14 | 2578 | 1 | 0.472034 |
| 15 | 11924 | 2 | 0.339169 |
| 16 | 7432 | 1 | 0.415764 |
| 17 | 8661 | 1 | 0.380422 |
| 18 | 11064 | 2 | 0.328703 |
| 19 | 7786 | 3 | 0.349086 |
| 20 | 2663 | 1 | 0.445664 |
| 21 | 10200 | 2 | 0.29449 |
| 22 | 10363 | 2 | 0.346697 |
| 23 | 8564 | 1 | 0.371122 |
| 24 | 10984 | 2 | 0.275269 |
| 25 | 4976 | 1 | 0.366308 |
| 26 | 5351 | 1 | 0.358412 |
| 27 | 9881 | 2 | 0.301543 |
| 28 | 9497 | 2 | 0.278197 |
| 29 | 8667 | 1 | 0.374947 |
| 30 | 10143 | 2 | 0.308042 |
| 31 | 8365 | 2 | 0.352472 |
| 32 | 5648 | 1 | 0.378782 |
| 33 | 7709 | 2 | 0.285256 |
| 34 | 5470 | 2 | 0.537633 |
| 35 | 3890 | 0 | 0 |
| 36 | 7477 | 2 | 0.2945 |
| 37 | 4784 | 3 | 0.438118 |
| 38 | 5425 | 0 | 0 |
| 39 | 7616 | 2 | 0.307646 |
| 40 | 8527 | 2 | 0.30626 |
| 41 | 2812 | 1 | 0.431315 |
| 42 | 4992 | 2 | 0.266187 |
| 43 | 7901 | 2 | 0.269329 |
| 44 | 4755 | 0 | 0 |
| 45 | 2210 | 0 | 0 |
| 46 | 10965 | 2 | 0.302466 |
| 47 | 11632 | 2 | 0.318209 |
| 48 | 10154 | 1 | 0.355027 |
| 49 | 9809 | 1 | 0.340951 |
| 50 | 0 | 0 | 0 |

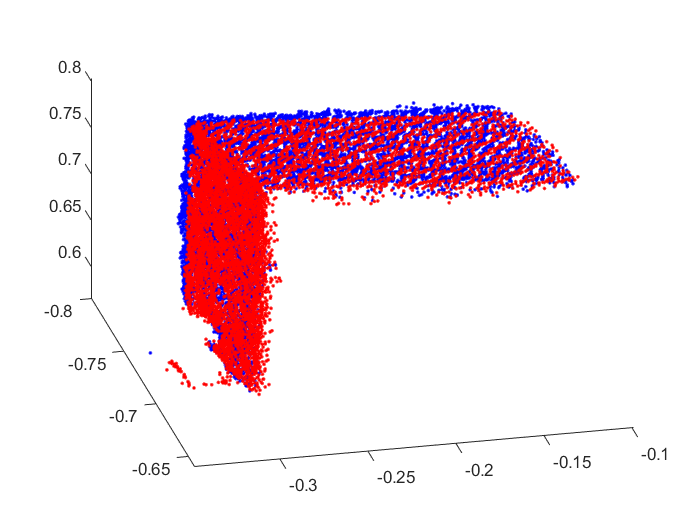
1. Estimate the 3D positions of the corners where planes meet



Corner1 = [-0.6370 -0.3418 0.8267];

Corner2 = [-0.7932 -0.2892 0.7798];

1. Estimate the rotation and translation



From frame 40 to frame 39:

R = [0.9985 -0.0274 -0.0478;

0.0144 0.9670 -0.2544;

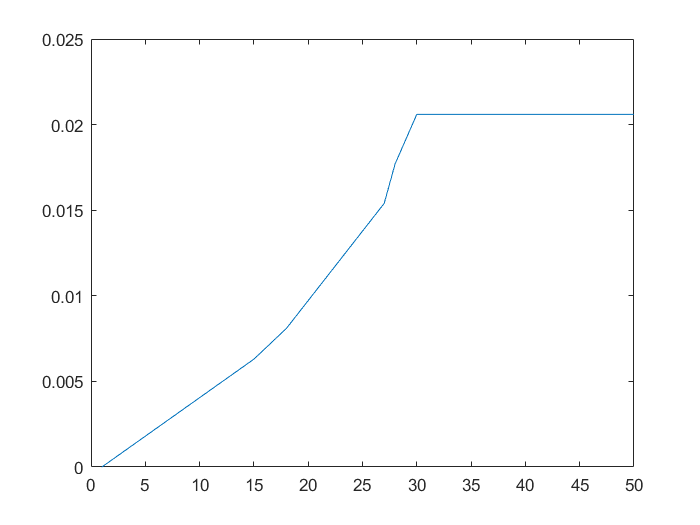
0.0532 0.2533 0.9659];

T = [-0.0258 -0.1992 -0.0907];

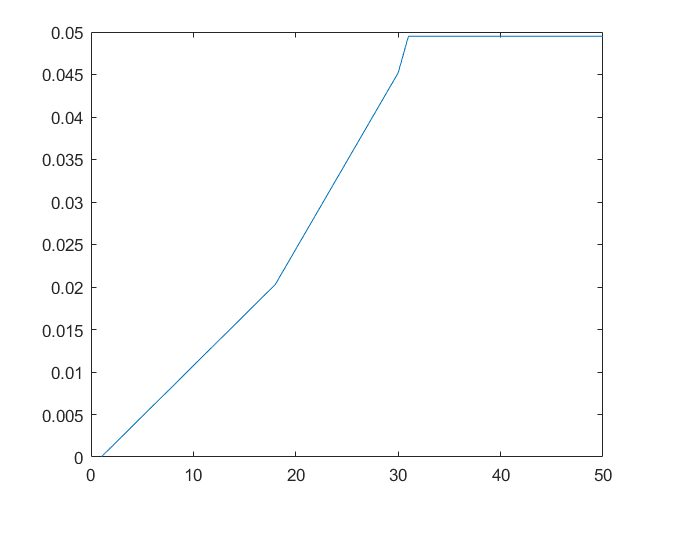
P\_40in39 = P\_40\*R+T;

1. Data fusing

Pick one of the large box surface planes visible in the first frame: (the only surface that have blue color)



Pick one of the corners visible in the first frame: (the up right corner of the surface with blue color in frame 6)



Discussion

By applying several different ways to remove noises, we actually have got a more clearer version of point cloud. But there are still some noises around the side plane, as shown in the figure below. The noise signal over there are quiet intensive such that K-neighbor algorithm cannot remove it. The noise is probably caused by the larger angle of the surface. Most of the infrared light signal just had just bounced away and only a few data is received by Kinect sensor.

Future work

To make the work easier, we can use more artificial method to complete the fusing process.

Conclusion:

The noise level of the point clouds after denoising is acceptable. We have good result for fusing all the data together. From that image, all the outer surfaces are perpendicular and the error of surface normal is 0.0206, which means our rotation matrix is good. But the translation matrix is not as good as the rotation matrix, with 0.0495 error in corner detection in the model evaluation. And we get on the RMS test, which tells us the plane is well fitted.