NPM3D - TP 5: Modeling

Adrien Golebiewski

IASD 2022-23

1 Question 1

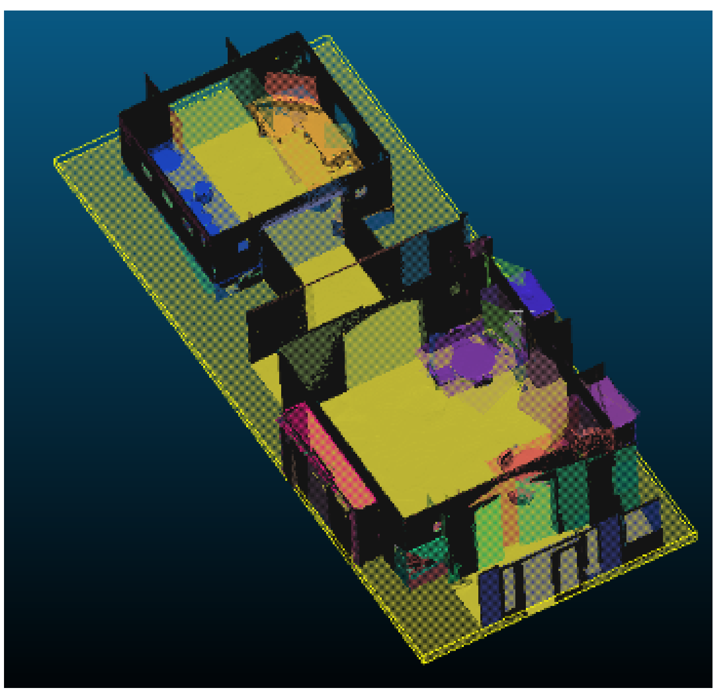


Figure 1: RANSAC plane detection (all planes)

We use the RANSAC plugin of CloudCompare to extract the main planes in the point cloud indoorscan.ply. In Fig.1, we add iteratively the most important planes that compose the point cloud,sorted by size. We use a minimum number of points per plane equal to 1500. We find that the floor in the point cloud is the main plane with an approximation of 900 000 points. We set a max distance to primitive of 0.1 and kept the default sampling resolution of 0.201.

Finnaly, I kept the value of 1 percent because I found that increasing the overlooking probability could lead to finding the planes in the wrong order, with smaller planes being found before bigger planes.

The screenshot above shows the "best" segmentation I obtained using CloudCompare.

2 Implement RANSAC in Python

3 Question 2

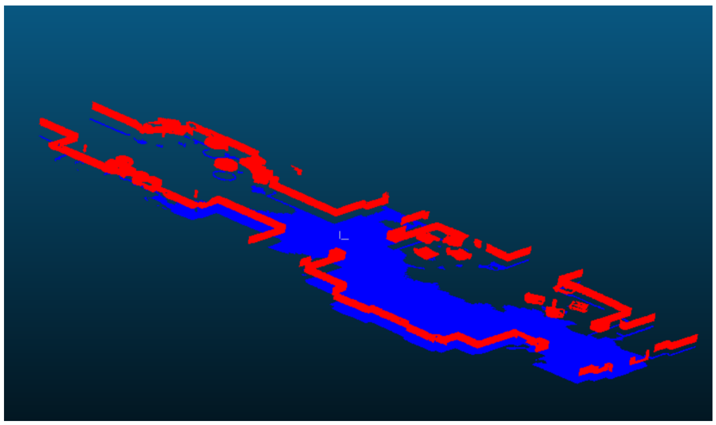


Figure 2: Two planes extracted consecutively by RANSAC

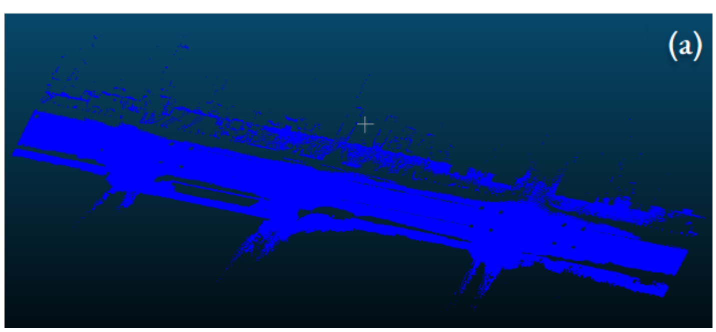
We can see that some of the extracted planes are not continuous and contain points which are far from each other. Since we do not use the spatial information of points (how there are distributed).

The RANSAC algorithm **doesn't take into account the proximity between the points** : two similar planes at the opposite of each other in the point cloud will be considered as belongin to the same plane. We should take into account the contiguity of the planes we are detecting.

4 Question 3

I choose the Notre-Dame-des-Champs pointcloud that we have seen in teh previous TP.

This point cloud describes a more complex scene with more planes than the previous ply file. In this context, the probability of drawing 3 points in the same plane is smaller. To tackle this issue, we increased the number of draws from 100 to 500.



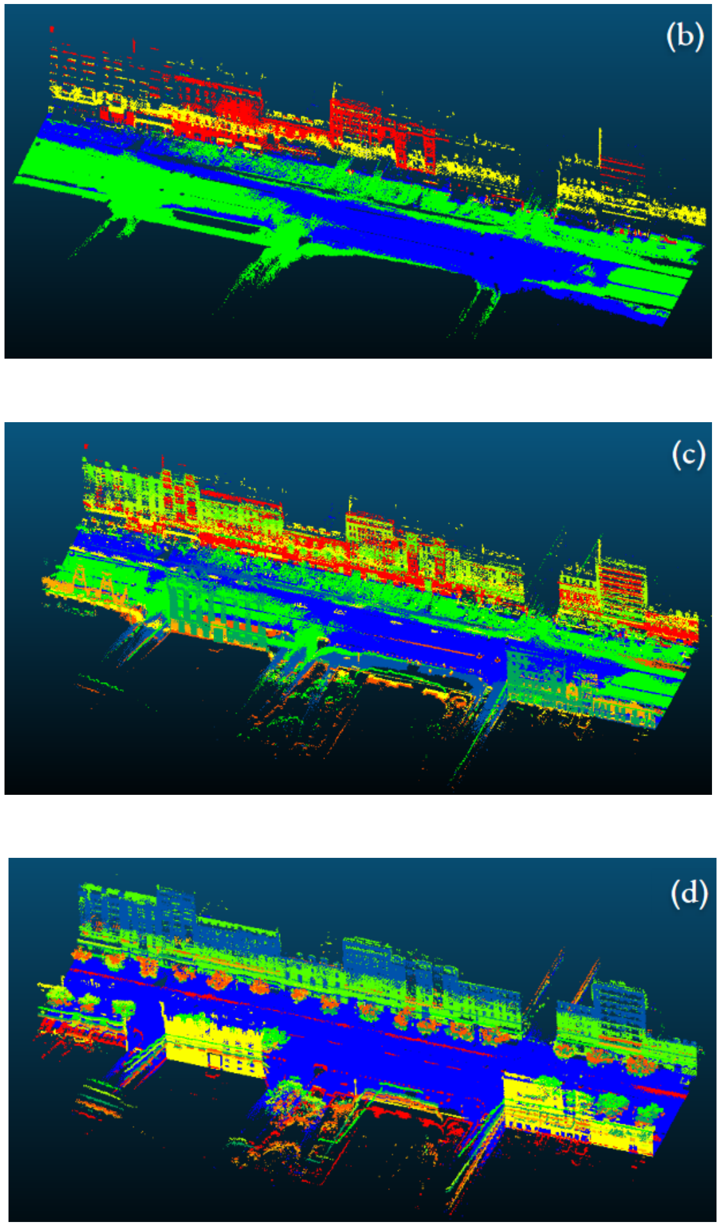


Figure 3: Extracting planes from NotreDameDesChamps1.ply using RANSAC (a) 1st plane, (b) 4 planes (c) 10 planes

We ran the RANSAC algorithm on **NotreDameDesChamps.ply** first using the same threshold in as before (0.1) but we as we can see in fig.4b-c, some of the planes overlap (facade of the building, road).

If the threshold is too low, as the point cloud is noisy, we can't detect properly the planes.

To prevent this behavior, we increased the threshold to 0.2 to obtain planes with more points. This parameter must be chosen according to the order of magnitude of the distance between the points of the cloud and according to how noisy the cloud is. But we still encounter the same problem as before with some planes (plane in orange and red in fig.4d) where the variance of points is high.

5 Question 4

The recursive RANSAC with normals function imply to include in the voting strategy of RANSAC, a threshold on the angle formed between the normal and the normal of the plane to fit at a given pint.

We have choosen the lowest value that would still count the floor as a single plane in order to set the threshold on the angle, The method with nb planes equal to 1 and increasing values of the maximum angle was run : a value of 0.1 radians was found. We kept the value of threshold on the distance to the plane of 0.1 :

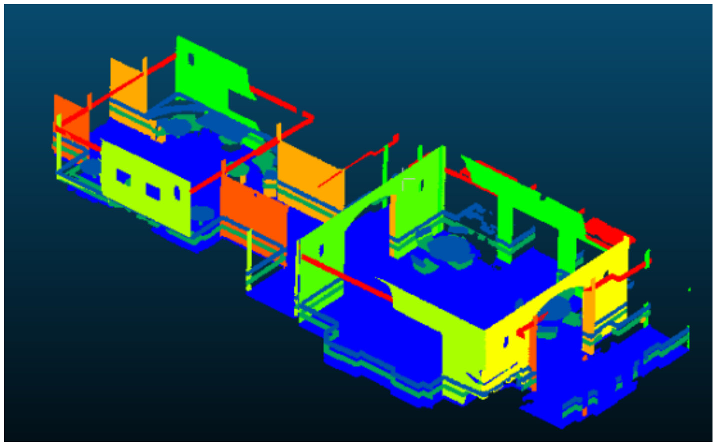


Figure 4: First five planes extracted using RANSAC

6 Bonus

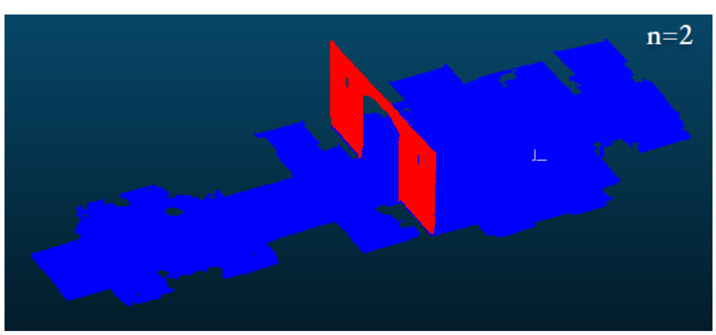
The tested method consists in changing the sampling method of triplets of points. During sampling, if one of the three points belongs to a wall opposite to the others, we will obtain the bad planes that we observed in our first implementation of RANSAC.

To tackle this limitation, we can sample a triplet of points that are close enough to avoid this problem but not too close either to avoid stability problems when computing the plane. Intuitively, we will first sample a single point and then compute its (k=2000) neighborhood using kdtree. We restrict this neighborhood set by taking the furthest half of this set and then we sample the 2 remaining points from this set of 1000 points.

Moreover, to reduce the within points distance for each computed plane, we tried first to discard planes that have a mean distance to the centroid of the plane greater than a threshold.

In this context, for each computed plane, we randomly sample (n=50) points from it and search for their nearest neighbor in a spherical neighborhood. The mean number of neighbors (over the n=50 points) can be used as a metric that measure the density of points in a plane.

By combining these 2 propositions, we recovered the main planes of the point cloud (floor and walls) :



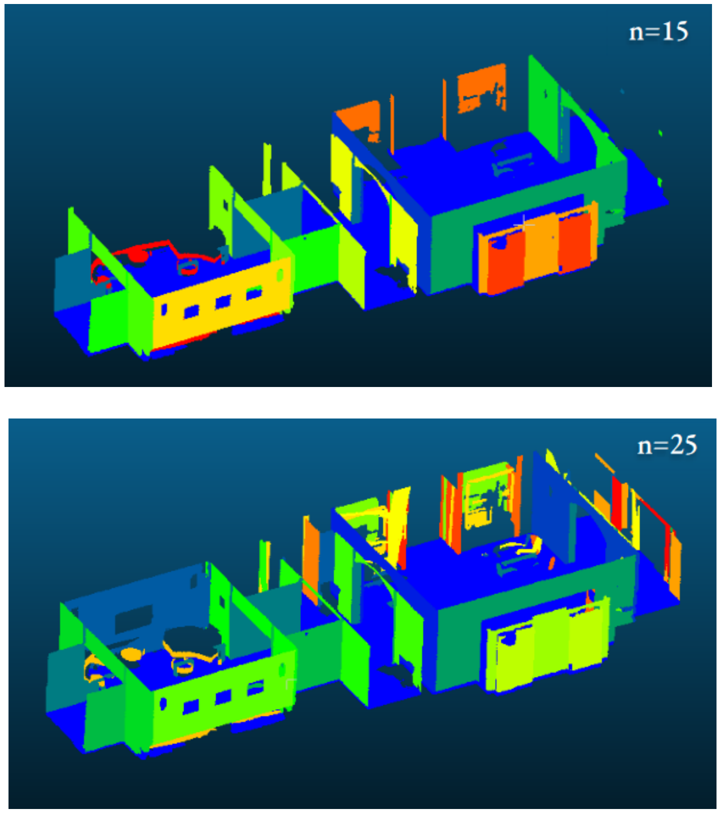


Figure 5: Improved RANSAC results

We can see that the results above **are very close to CloudCompare RANSAC plugin**.