Exercise 5: Plane fitting using RANSAC and MLESAC

Max Tamussino, 01611815

December 21, 2020

Contents

1   Single plane detection

The RANSAC algorithm is used for the detection of planes in three-dimensional image data, so-called pointclouds. Figure 1 shows the detected plane in an example pointcloud containing one main plane and several objects on top of it. The used parameters are discussed in the following Section.
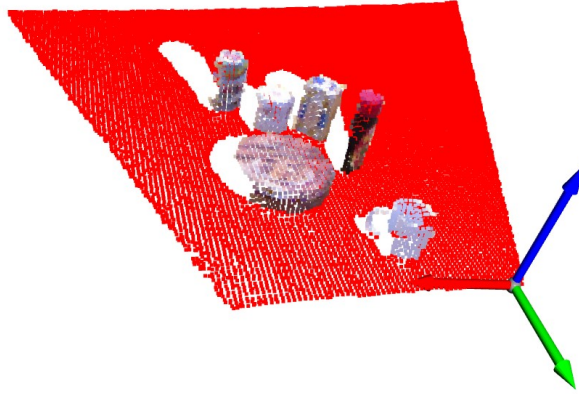

Figure 1: Detected plane (red) in simple pointcloud
$t_I = 0.02$, $d_S = 0.8$, $c = 0.9$

2   Parameter influence

For this experiment, three parameters of the RANSAC algorithm were varied and examined for their influence on the result: The inlier threshold t_I, the minimum distance between samples d_S and the termination confidence c. The previously used parameters in Figure 1 were found to yield appropriate results and were therefore used as the base values.

The parameter t_I significantly influences the algorithm. Large values lead to the inclusion of parts of objects into the calculated plane, while small values lead to the incomplete detection of the plane points. The latter is depicted in Figure 2a and could lead to multiple detections of the same plane, however objects on the plane remain untouched. Generally, this parameter should be fitted to the expected noise level to optimally include all valid data points. The other parameters seem to have no or little effect in the examined scenario, as visible in Figures 2b and 2c. Planes, which result from closely selected points (which is possible for low d_S), are more than likely to be removed by the following RANSAC iteration, especially in this nearly ideal scenario. The number of inliers for the correct plane is just too high. The scenario also leads to the insignificance of c, as the first iterations will yield high confidence values, which will lie over this threshold. These parameters should be observed in other environments to examine their influence.

3   Multi plane extraction

The multi-plane extraction using RANSAC was performed on three different scenes, all depicted in Figure 3. The results are the most accurate in Figure 3b, as the main few

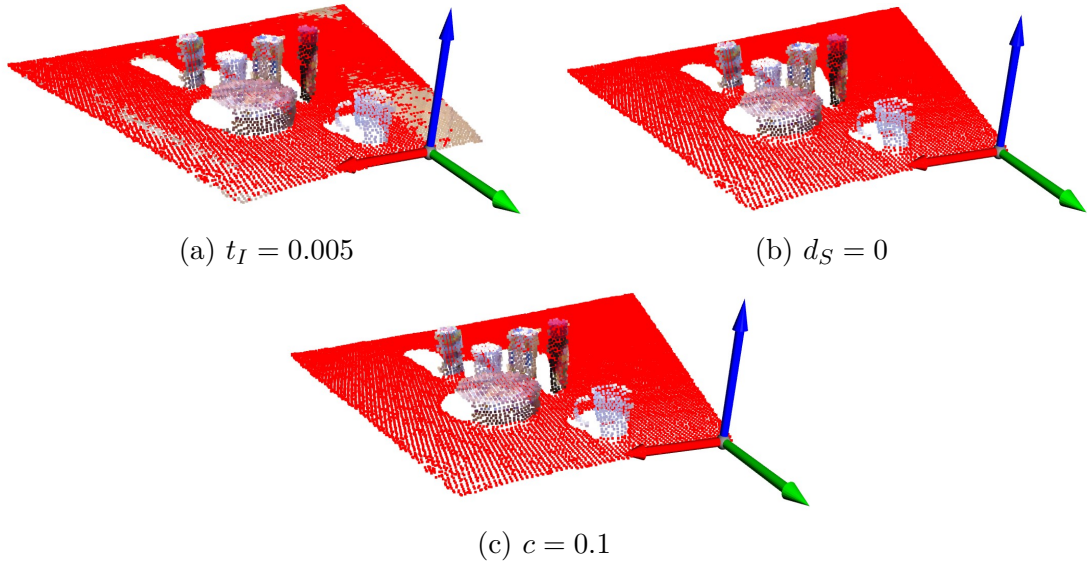(a) $t_I = 0.005$

(b) $d_S = 0$

(c) $c = 0.1$

Figure 2: Results of the plane detection with varied RANSAC parameters

planes are detected with only one additional insignificant plane. Figure 3c shows a higher number of additional planes which were erroneously extracted. Even the main kitchen front was not completely extracted as one plane. Figure 3a shows the difficulty of the random starting point selection: The two monitors are wrongly detected to be one plane. The main desk plane is however detected correctly and completely.

In addition, the influence of different downsampling strategies was tested, especially of uniform and voxel grid downsampling. Uniform downsampling in Figure 4a shows vertical lines of pixels in the lower half, possibly because the number of pixels per row is a multiple of ten. This effect does not occur using voxel downsampling, as shown in Figure 4b: The pixels are perfectly uniformly distributed. This technique is however computationally more expensive.
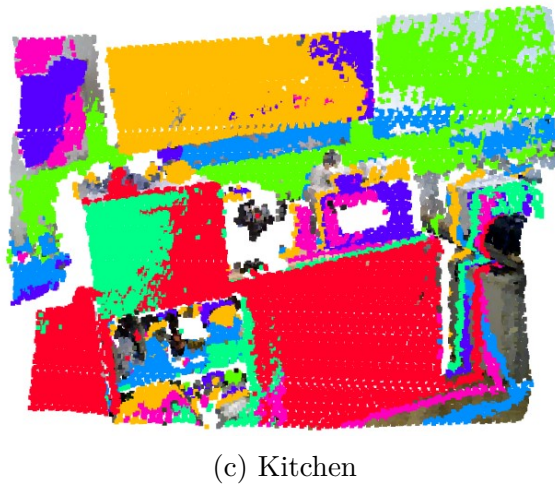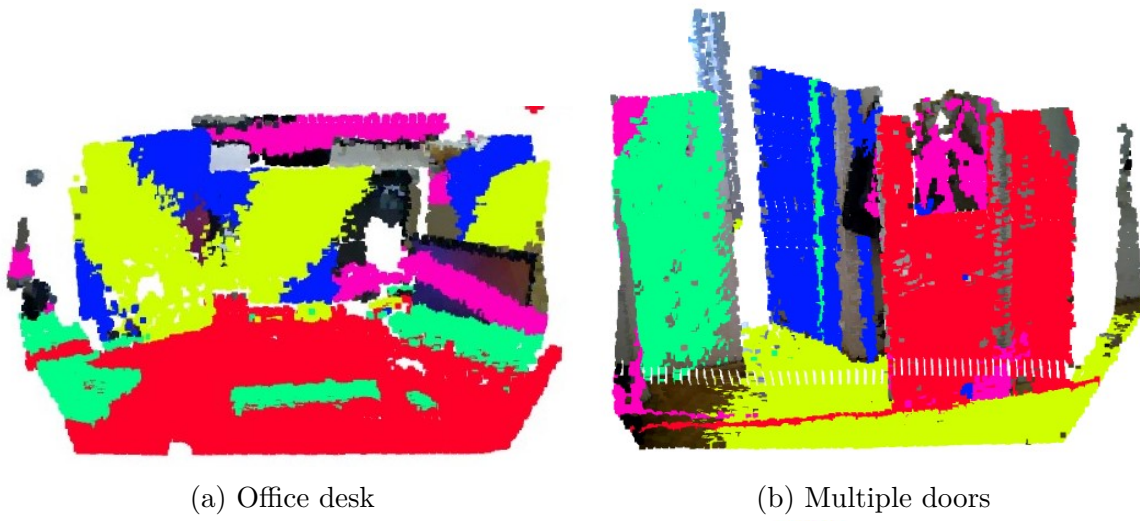
(a) Office desk

(b) Multiple doors

(c) Kitchen

Figure 3: Result of repeated RANSAC plane detection on different scenes

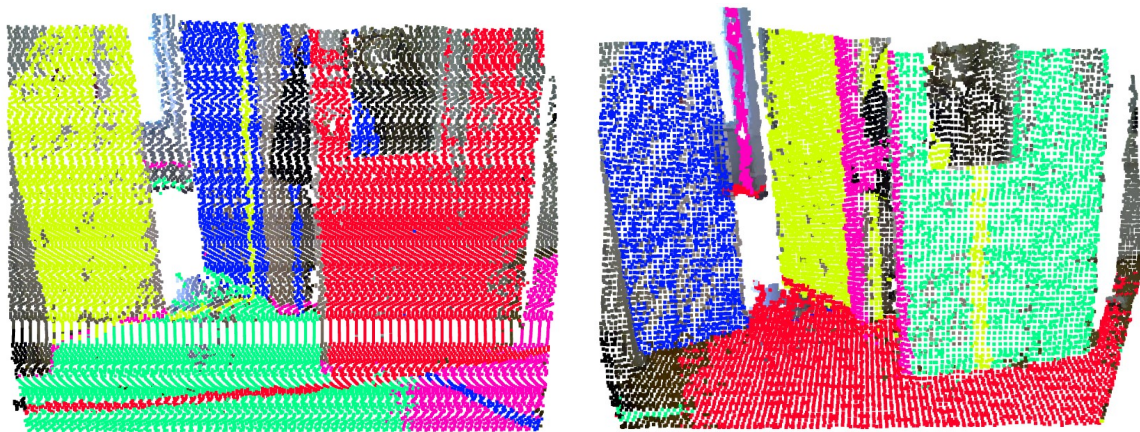(a) Uniform downsampling (every tenth point) (b) Voxel downsampling (using voxel size 0.03)

Figure 4: Plane extraction using different downsampling techniques

4 Performance comparison: RANSAC, MSAC and MLESAC

For this experiment, a noisy plane was created and additional noise pixels were added. RANSAC, MSAC and MLESAC were applied to this dataset 500 times and the performance statistics were compared.

4.1 Angle error

Firstly, the error of the resulting angle is examined. Figure 5 shows error statistics for all three techniques. The superiority of MLESAC is clearly visible, as its mean error is significantly lower. MSACs mean error is a little higher than RANSACs, however its maximum generated error is lower - it depends on the application which one of those two will be considered more useful.
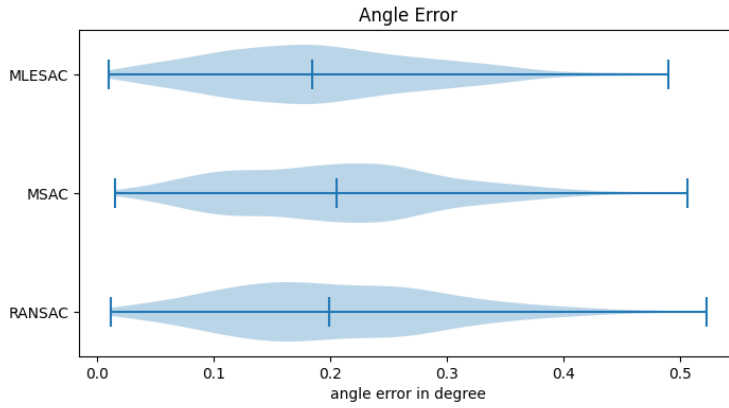

Figure 5: Error statistics of the detected angle for RANSAC, MSAC and MLESAC

4.2 Number of iterations

Figure 6 shows box-plots of the needed number of iterations until the algorithm terminates. As in the previous experiment, a performance benefit of MLESAC is visible. The median number of iterations is lower for this technique. MSAC and RANSAC show very similar iteration number statistics, which is surprising as a performance benefit of the quadratic error function was expected.
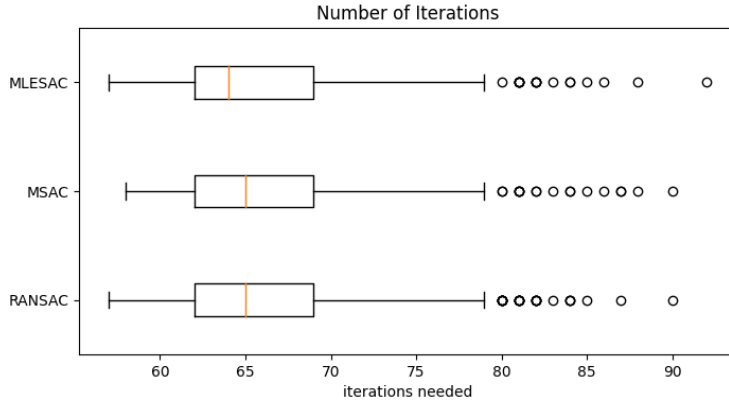

Figure 6: Iteration number statistics for RANSAC, MSAC and MLESAC