PLANE DETECTION IN POINT CLOUDS VIA RANSAC

D. Kuramin

TU Berlin, Institute of Geodesy and Geoinformation Science, Berlin, Germany
kuramin@campus.tu-berlin.de

KEY WORDS: Plane detection, model fitting, point cloud, laser scanning, computational geometry, 3D modelling

ABSTRACT:

A problem of fitting a model to experimental data is one of the main tasks in science. Given a dataset which is results of measurements of some properties of real objects, scientific approach is able to detect and to quantify relationships between these properties which means to describe objects of the real world. Such research requires some knowledge: which kind of model is supposed to describe these objects, which kind of relationships are supposed to be present in the given dataset. Using this basic information, algorithms of data analysis are able to estimate values of model parameters. Algorithms differ from each other in many aspects, one of them is treatment of erroneous measurements. There are algorithms which are efficient, even if unreliable measurements make up a significant part of a dataset, and are able to detect and remove such measurements. These algorithms are very useful for 3D modelling based on results of laser scanning, which are big point clouds with a large amount of abundant members. One of these algorithms is Random Sample Consensus (RANSAC). The most useful application of this algorithm for laser scanning point clouds is detection of planes, which is crucial for determination and representation of buildings. The current paper describes an application of RANSAC for the task of plane detection in a point cloud and discusses results of its practical implementation.

1. INTRODUCTION

Classical algorithms of parameter estimation (e.g., Least Squares Adjustment) find the best model parameters using all the presented data. As a result, gross errors make an impact on the calculated model parameters. Even if some techniques (e.g. Data Snooping (Baarda, 1968)) are able to identify outliers in the dataset by statistical approach and eliminate them as it is shown in the paper of (Schwarz and Kok, 1993), this process will not succeed in case of a significant percentage of outliers in input data (more than 5%).
Another possible approach would be an application of Robust Parameter Estimation as it is done in the article of (Krarup et al., 1980): after the conventional least squares adjustment weights of measurements are updated based on values of residuals and new least squares adjustment is applied. With every iteration until convergence weights of gross errors become smaller and smaller, so they have an infinitesimal impact on the final result. However, this method also has a disadvantage: it's complexity and computational cost are very high.
The approach suggested in the current article uses a technique which is opposite to the aforementioned methods: instead of iterating through derivation of a solution from as much of the data as possible and elimination of outliers, RANSAC uses the smallest possible dataset to define the model and then computes an amount of data which agree with it. In this case, gross errors don't have any influence: the only amount of consistent data plays a role.
In case of an application for plane detection, definition of a model consists in a definition of coefficients of the plane equation, which are dependent on the coordinates of three points from the input point cloud. The consistency of a single point with the model is calculated from the perpendicular distance between the point and the plane: if the distance is smaller than some predefined threshold, then the point is consistent. Further description of definition and treatment of compatible points is given later in the current article.

2. RANSAC ALGORITHM FOR DETECTION OF PLANES IN A POINTCLOUD

The RANSAC algorithm of plane detection finds many applications, one of them is processing of the results of terrestrial laser scanning in order to find a ground surface, walls, and roofs of the buildings for the goals of 3D modelling. Thus, the input point cloud usually represents not one, but many planes, all of them can be rectified by the suggested method. Every found plane together with consistent points should be deleted from the point cloud, then other planes acquire the possibility to be detected. Moreover, RANSAC algorithm requires values of several parameters to be predefined. Parameters $resid$, $iter$, $area$, min_cons, $found$ or $rest$ are called hyper-parameters of the algorithm and they define precision and quantity of found planes. The algorithm of RANSAC for plane detection looks as following:

- Random triplet of points M_1, M_2, M_3 is chosen from the initial input point cloud. If three points are not collinear, they define a plane. Since dependency of coefficients A, B, C, D of plane equation $Ax + By + Cz + D = 0$ from coordinates of points $M_1(x_1, y_1, z_1), M_2(x_2, y_2, z_2), M_3(x_3, y_3, z_3)$ is defined by equation (1):

$$\begin{bmatrix} x - x_1 & y - y_1 & z - z_1 \\ x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_1 & y_3 - y_1 & z_3 - z_1 \end{bmatrix} = 0 \quad (1)$$

the formulas for coefficients are the following:

$$\begin{aligned} A &= y_1x_2 + y_3z_1 + y_2z_3 - y_3z_2 - y_2z_1 - y_1z_3 \\ B &= x_3z_2 + x_2z_1 + x_1z_3 - x_1z_2 - x_3z_1 - x_2z_3 \\ C &= x_1y_2 + x_3y_1 + x_2y_3 - x_3y_2 - x_2y_1 - x_1y_3 \\ D &= x_3y_2z_1 + x_2y_1z_3 + x_1y_3z_2 - \\ &- x_1y_2z_3 - x_3y_1z_2 - x_2y_3z_1 \end{aligned} \quad (2)$$

- Now distance from every point $P(x_p, y_p, z_p)$ of the cloud to this plane can be calculated by the following formula:

$$d = \frac{Ax_p + By_p + Cz_p + D}{\sqrt{A^2 + B^2 + C^2}} \quad (3)$$

- Points which are close enough to the plane (i.e. distance is lower than hyper-parameter $resid$) are called "consensus set" of this plane. Points of triplet also are members of the consensus set.

- If the size of consensus set is more than hyper-parameter min_cons, the plane is considered being the best fit for the current point cloud. If consensus set is not big enough, new triplet must be found and a new consensus set calculated. If a search for the suitable consensus set doesn't succeed for $iter$ times, the biggest of them is taken as acceptable.

- Found suitable consensus set means that definition of the plane has succeeded. Depending on the type of required result, either plane equation coefficients or consensus set itself or even both are saved as a new member of resulting vector.

- In order to find the next plane, points of a consensus set of the current step must be deleted from the point cloud. Then the next iteration starts from the beginning, using the resting part of the point cloud as input dataset.

- Iterating process continues until the predefined number of planes $found$ is found or while a size of the resting point cloud is more than $rest$. The detailed description of values of hyper-parameters will be given further.

Check on the eligibility of triplet after the random choice of points is performed based on the area of the triangle created by these points. Triplet is considered being eligible, if the area is bigger than hyper-parameter $area$. This fact means that 2 criterions are satisfied: points are not collinear and points are not too close to each other. Triplets of points which are distant from each other tend to define more acceptable planes than points located densely. Thus, hyper-parameters which define the following aspects of RANSAC are:

- $area$ - defines eligibility condition for triplets of points;

- $resid$ - sets how far can point of consensus set be from the plane;

- min_cons - corresponds to a minimal size of acceptable consensus set;

- $iter$ - is a number of iterations performed in order to find an acceptable consensus set for certain point cloud;

- $found$ and $rest$ - define termination condition of planes detection loop. Since $found$ is a number of found planes and $rest$ corresponds to a maximal size of the resting point cloud, these parameters are mutually exclusive and only one of them must be used. The logic of choice of hyper-parameters and their values is provided in the next chapter.

The result of RANSAC application is the vector of datasets which are consensus sets of corresponding planes. The last member of this vector is set of points which were not compatible with any plane. Possible improvement of RANSAC algorithm can be done here by application of Least Squares Adjustment: if the output of algorithm contains both vector of plane coefficients and vector of corresponding consensus sets, coefficients of planes can be recalculated using corresponding consensus sets. To do so, all members of consensus set are used as input for least squares adjustment, already known coefficients of the plane can be utilized as initial values of unknowns. Such improvement has an advantage, because of the fact that recalculated plane coefficients are not based on coordinates of only three points from consensus set, but are influenced by every member of consensus set (Fischler and Bolles, 1981).
Another way of improvement of RANSAC is a usage of octree structure as a storage of the point cloud, which is described in the papers of (Meagher, 1982) and (Schnabel et al., 2007). The main idea of an octree is partitioning of three-dimensional space by recursive subdivision of it into eight octants and building of a corresponding tree structure depending on the presence of points in octants. As a result, location of every point is described by position of its leaf in the octree and distance between points can be assessed as distance between corresponding octants, which is efficiently calculated from relative path from one leaf to another. For application of RANSAC it means that points which belong to octants distant from the current plane can be excluded from potential members of the consensus set of this plane. Such approach speeds up the calculation and is very efficient for extra-large datasets.

3. PRACTICAL IMPLEMENTATION OF RANSAC

The algorithm of RANSAC was applied to detect planes in the point cloud, which is the result of laser scanning of buildings in Charite clinic in Berlin. This dataset contains almost 400 000 points, some part of them has values of all coordinates and intensity equal to zero. The reason of presence of such points in the cloud is described in the article (Kuramin, 2018). These points with zero values are useless for plane detection and they must be deleted from the cloud before application of RANSAC.
The calculation was performed for different values of hyper-parameters, only value of triplet eligibility condition was fixed as $area = 0.5m^2$, because it's changing inside of a reasonable range does not have any noticeable influence on the result. Among several options for the termination criterion of the plane detection loop, the size $rest$ of the resting point cloud was chosen, because it can be defined as a function of the size of the initial point cloud. Another option to stop the process of looking for planes is based on the number $found$ of found planes, but its appropriate value is rather hard to predict. On the other hand, size of the resting point cloud can be set relative to the size of the initial point cloud (i.e. 10%), so that small planar segments will not be found, but main planes will be detected regardless to the size of the whole dataset.
After planar segments were detected, each of them got its own color. Assignment of colors was based on the idea to provide the biggest planar segments with colors which are the most distinguishable from each other. Those points which are situated sporadically or belong to some non-planar objects, like a body of a person, are not compatible with any plane and were colorized in black. The application shows the number of planar segments and size of each segment, as well as the number of resting points.
The worst result was achieved using too low values of $iter = 0.01$ and $min\_cons = 0.1$: in this case number of efforts to detect every new plane is equal to 1% of a size of a resting point cloud and plane is considered found if only 10% of resting cloud is compatible with it. Such values lead to incorrect detection of planes, which is shown on Figure 1:

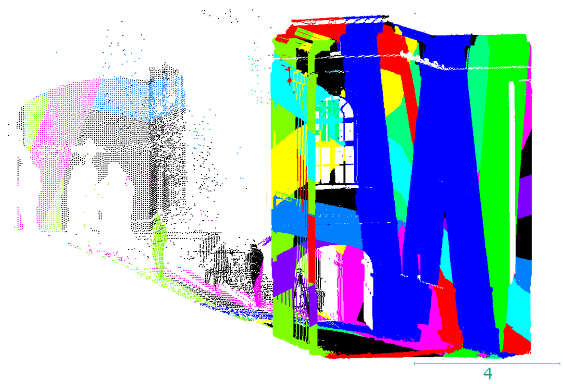

Figure 1. Result of RANSAC with hyper-parameter values $resid = 0.5$, $iter = 0.01$, $rest = 0.1$, $min\_cons = 0.1$

Figure 2 shows the result of RANSAC using more appropriate values of $iter = 0.1$ and $min\_cons = 1.0$. Here planes are detected better, in fact such high value of min_cons forces to check all

$iter$ planes and to choose the best among them. But still, many points near plane intersections get wrong assignment because of too high value of the buffer width $resid = 0.5$. Moreover, too many points are black, because value of $rest = 0.1$ terminates plane search when 10% of the cloud remains unassigned.

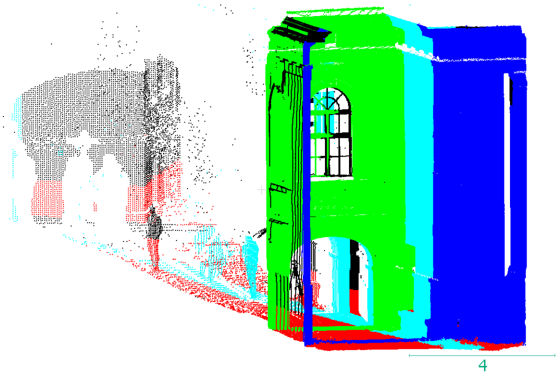

Figure 2. Result of RANSAC with hyper-parameter values $resid = 0.5$, $iter = 0.1$, $rest = 0.1$, $min\_cons = 1.0$

The result of processing using value of $resid = 0.02m$ is shown on the Figure 3. Such a small value of $resid$ leads to dissociation of big almost planar segments to many small parts. As a result, output is too detailed, has too much segments. Also this case uses value of $rest = 0.01$, which leaves less black points and value of $min\_cons = 0.7$, which decreases time of definition of every plane.

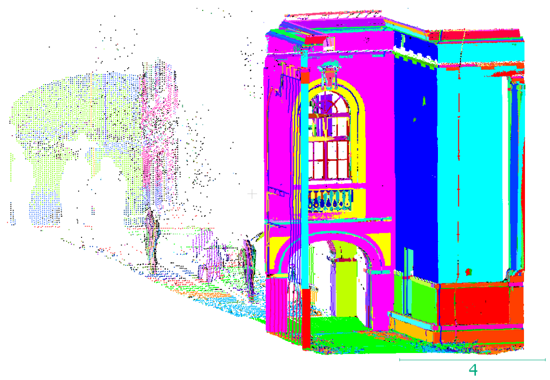

Figure 3. Result of RANSAC with hyper-parameter values $resid = 0.02$, $iter = 0.1$, $rest = 0.01$, $min\_cons = 0.7$

The best result was obtained with following values of hyper-parameters: $resid = 0.1$, $iter = 0.1$, $rest = 0.01$, $min\_cons = 0.4$. In this case there are not so many unassigned points (3098 points), assignments are correct, every wall is determined as a single plane and number of planes is not too high (22 planes). This result is illustrated by the Figure 4:

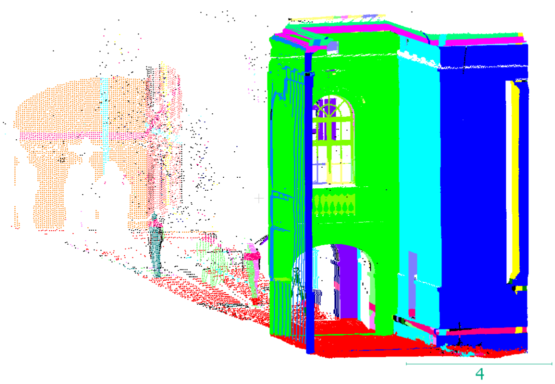

Figure 4. Result of RANSAC with hyper-parameter values $resid = 0.1$, $iter = 0.1$, $rest = 0.01$, $min\_cons = 0.4$

4. CONCLUSION

The current paper described an application of RANSAC algorithm for plane detection in the point cloud. Acquired results are high qualified: segments are planar within a reasonable tolerance.

The task of planar shapes detection can be solved by different algorithms. Competitors of RANSAC, such as techniques of Data Snooping and Robust Parameter Estimation, were considered and found being less advisable. However, one more method alternative to RANSAC must be mentioned. This method is Hough transformation. The article of (Tarsha-Kurdi et al., 2007) suggests an extension of traditional two-dimensional Hough transformation (Hough, 1962) for the three-dimensional case: every point of Euclidean space can be represented as a sinusoidal surface in parametric space. Then the intersection of many sinusoidal surfaces means that all corresponding points belong to the same plane. But practical implementation of this approach has shown that it's less efficient than RANSAC (Tarsha-Kurdi et al., 2007).

Also, further improvement of RANSAC algorithm can be done by application of Least Squares Adjustment or by usage of octree structure as a storage of the point cloud. Such improvement speeds up the calculation and makes RANSAC much more preferable than its competitors.

REFERENCES

Baarda, W., 1968. A testing procedure for use in geodetic networks. *Publications of Netherlands Geodetic Commission.*

Fischler, M. and Bolles, R., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM,* 24(6), pp. 381–395.

Hough, P. V. C., 1962. Method and means for recognizing complex patterns. *U.S. patent 3.069.654.*

Krarup, T., Juhl, J. and Kubik, K., 1980. Gotterdammerung over least squares adjustment. *Proceedings of 14th Congress of the International Society of Photogrammetry,* pp. 369–378.

Kuramin, D., 2018. Algorithm of intensity based registration for terrestrial laser scans,. *Proceedings of Laser Scanning Workshop in the Technical University of Berlin.*

Meagher, D., 1982. Geometric modelling using octree encoding. *Computer Graphics and Image Processing,* 19(6), pp. 129–147.

Schnabel, R., Wahl, R. and Klein, R., 2007. Efficient RANSAC for point cloud shape detection. *Computer Graphics Forum,* 26(2), pp. 214–226.

Schwarz, C. R. and Kok, J., 1993. Blunder detection and data snooping in LS and robust adjustment. *Journal of surveying engineering,* 119(14), pp. 127–146.

Tarsha-Kurdi, F., Landes, T. and Grussenmeyer, P., 2007. Hough-Transform and Extended RANSAC algorithms for automatic detection of 3D building roof planes from lidar data. *Proceedings of ISPRS Workshop on Laser Scanning 2007 and SilviLaser,* 36, pp. 407–412.