# AN ORIGINAL ALGORITHM FOR BIM GENERATION FROM INDOOR SURVEY POINT CLOUDS

F. Capocchiano [1, 2], R. Ravanelli [2 *]

[1] Sapienza School for Advanced Studies, Rome, Italy
capocchiano.1744374@studenti.uniroma1.it
[2] Geodesy and Geomatics Division, DICEA - Sapienza University of Rome, Rome, Italy
roberta.ravanelli@uniroma1.it

**Commission IV, WG IV/5**

**KEY WORDS:** Building Information Modeling (BIM), scan to BIM, point cloud, indoor mapping

**ABSTRACT:**

Nowadays, it is essential to find new strategies, able to perform the first step of the scan-to-BIM process, by retrieving the geometrical information contained in point clouds that are so easily collected through laser scanners and range cameras. This paper presents a new algorithm for the automatic extraction of the layout and the height of a small indoor environment from its point cloud. In particular, the algorithm was tested on a point cloud of 600000 vertices, selected from the dataset of the ISPRS benchmark on indoor modelling. The preliminary results are encouraging: the 3D shape (layout and height) of the investigated room is effectively reconstructed.

## 1 INTRODUCTION

Building Information Modelling (BIM) offers several benefits in the design, planning, and construction of new buildings as well as in the management and maintenance of the existing ones. Frequently, however, BIMs are not available for older constructions, and, in some cases, even blueprints are missing. Consequently, it is essential to find new strategies to efficiently generate BIMs for the already existing and used buildings (Xiong et al., 2013). At the same time, the recent advances in surveying technology (e.g. laser scanners and range cameras) make relatively easy to collect dense point clouds of indoor environments, even if the current processing solutions are not yet ripe to analyse the massive datasets produced. Therefore, it is increasingly necessary to develop automatic and efficient algorithms able to rapidly extract at least the geometric BIM features from indoor point clouds, such as layouts and room heights.

In this paper, the algorithm presented in (Capocchiano et al., 2017) is extended and enhanced. Originally designed to extract layouts from the 3D models of rooms acquired by means of a low-cost range camera, the algorithm is now able to process also very dense point clouds provided by professional laser scanners. Specifically, the implementation was refined in order to:

- increase the computational efficiency of the algorithm;
- retrieve also the height of the scanned room (supposed constant) along with the layout.

In particular, the new version was implemetned in Python and tested on a point cloud of 600000 vertices (Figure 1), selected from the dataset of the ISPRS benchmark on indoor modelling (Khoshelham et al., 2017).

This paper is organized as follows: Section 2 describes the workflow of the algorithm in detail and Section 3 presents and discusses the obtained results. Finally, in Section 4, some conclusions are drawn and future prospects are outlined.

*Corresponding author.

## 2 THE ALGORITHM

The algorithm retrieves the layout and the heigth of an indoor environment (e.g. a room) starting from its 3D point cloud, which have to include the ceiling, the upper portion of the walls and a small portion of the floor (Figure 1). If the mesh is also available, as is common when working with mobile devices equipped with range cameras, it could be useful to carry out a preliminary statistical analysis of the length of the polygon edges (for the selection of the triangulation parameters, as it will be described later). However, from this point on, only the information contained in the vertices of the point cloud are considered by the algorithm. In particular, the user has to set the values of different parameters belonging to the following three categories:

- parameters depending on the available computing power (e.g. maximum number of executions for a loop);
- parameters depending on the features of the specific sensor whereby the point cloud was collected;
- parameters strictly related to the specific survey, such as the number of the sides of the room.

The reference frame associated with the point cloud is assumed arbitrary (even though professional scanners include sensors capable to detect the vertical direction by means of gravity), thus the equation of the plane modeling the ceiling is identified through the RANdom SAmple Consensus (RANSAC) algorithm (Fischler and Bolles, 1981). RANSAC is iterated multiple times: each iteration takes as input the inlier set of the previous one (Ravanelli et al., 2015) and works with a threshold reduced by 1 cm with respect to the previous iteration. The parameters required are:

- the starting threshold, depending on the sensor accuracy and the real ceiling flatness;
- the stop criteria. In point clouds collected by low-cost range cameras, the ceiling may present a variable slope: an excessively small threshold would force RANSAC to restrict the

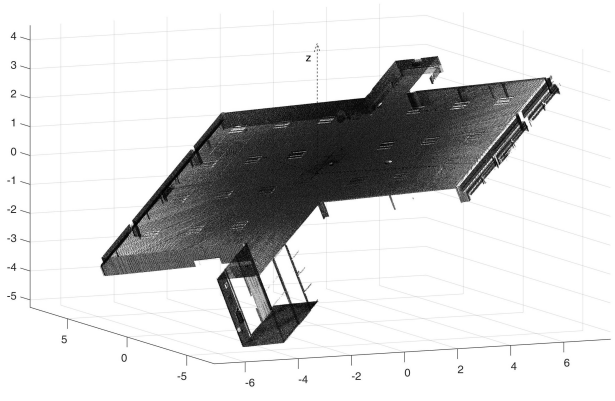


Figure 1: Input point cloud.

inlier set to a portion of the ceiling with uniform slope, i.e. invalidating the plane estimation. To prevent this issue, conditions on the minimum threshold, the maximum angle (the angle obtained by inverting the scalar product of the two normal unit vectors to the planes in question and is comprised between $0^\circ$ and $180^\circ$) between two consecutive estimated planes and the maximum reduction factor (the relationship between the number of inliers of the iteration N, and the number of inliers of the iteration N - 1 must be greater than a certain value, typically between 0.8 and 0.9) for the number of inliers, should be considered.

Eventually, the final RANSAC result is refined by estimating the plane parameters only on the inlier set through the Ordinary Least Squares method. After that, a new reference frame with the $Z$' axis orthogonal to the ceiling plane (pointing outward) is defined (Figure 2): the new coordinates of the vertices are subsequently computed by means of quaternions (Sansò, 1973).

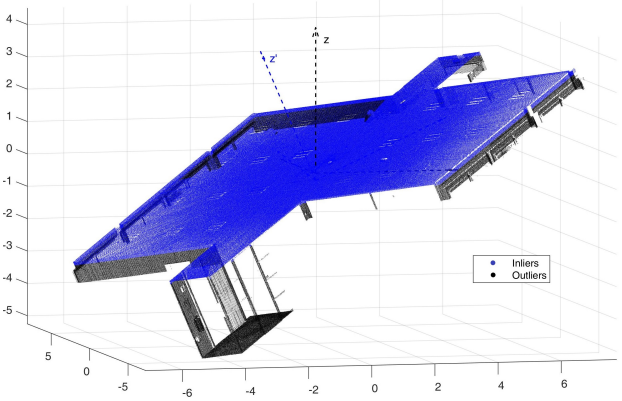



Figure 2: RANSAC plane estimation.

This reference frame transformation is useful because in this way the room height can be easily computed as the absolute value of the minimum of $Z$'(Figure 3). Subsequently, the portion of the point cloud exceeding a specific distance from the plane is excluded from further processing (Figure 3). This distance depends on the features of the environment surveyed, e.g. the presence of windows, false ceilings, furniture, air conditioning and in general every object that can obstruct the scanning process and hide parts of the ceiling-walls border. The remaining points are thus orthogonally projected on the ceiling plane, reducing the original 3-dimensional problem into a 2-dimensional one (Figure 4).

Then, considering that only the boundary of the point cloud is interesting for the perimeter detection, namely for the layout extraction, a cloud reduction is performed to keep an acceptable execution time in the following processing steps. This simplification is virtually mandatory when very dense point clouds collected by professional laser scanners are used and can be summarized in the following steps:

1. the bounding box of the 2D point cloud (from now on $P_0$) is subdivided into a grid of squares $s'_{ij}$ with side $l_s$;

2. the squares containing at least one point are included in the reduced point cloud $P_1$if they share at least one vertex with an empty square or if they are placed at the border of the bounding box; at this stage the final reduced point cloud $P_r$ is equal to $P_1$;

3. the squares containing at least one point previously excluded from $P_r$ and sharing at least one vertex with a square which belongs to $P_r$ form a "monitored area" $M_1$;

4. the bounding box of the full 2D point cloud is subdivided again into squares ($s''_{ij}$) with side length $l_s/k_1$ ($k_1 \in N$). A second reduced point cloud $P_2$ is assembled under the above-mentioned conditions;

5. if $s''_{ab} \subset P_2$, $s''_{ab} \subset M_1$ and $s''_{ab} \not\subset P_1$ then the $s''_{ab}$ is added to the final reduced point cloud $P_r$. If $s''_{ab} \subset P_1$ but $s''_{ab} \not\subset P_2$ then $s''_{ab}$ is removed from $P_r$. Whenever a square $s'_{cd}$ belonging to $M_1$ has a point added to the final reduced point cloud $P_r$, every square $s'_{ab}$ that share a vertex with such square is added to $M_1$, and the control operated at the beginning of this point is reiterated, until no more points are added to $P_r$;

6. it is possible to define a new monitored area $M_2$ (same criteria adopted for $M_1$, this time with $s''_{ij}$ squares) together with a new simplified point cloud $P_3$ comprised of squares $s'''_{ij}$ with side length $l_s/k_1k_2$ ($k_1, k_2 \in N$), in order to refine again $P_1$ following the procedure set up in the previous step.

In general, the procedure described in step 6 can ideally take place many times with an arbitrary number of reduced point clouds, but limiting the process to $P_2$, with a proper choice of the parameter $k_1$, allows to obtain satisfactory results. Moreover, it is highly recommended to avoid removing further points from the simplified cloud of the previous step (as explained in step 5) because of the excessive narrowing of the border area that would occur in this case. Regarding the choice of $l_s$, the task is complicated by conflicting needs. A small value of $l_s$ assures that all the border points required for the layout extraction are included in the simplified point cloud (Figure 5). In this way, though, the border of a hole placed in the central portion of the point cloud can be erroneously added to the reduced point cloud (Figure 5b). On the other hand, a large value of $l_s$ prevents the inclusion of central holes, but at the cost of a lower resolution in the border zone: significant points may be excluded if located near reflex interior angles (in room with concave geometry) and a large number of non-significant points can be included in the $s$' squares in $P_1$ together with border ones (Figure 6).

The aim of the monitored area is precisely to find a compromise between these two equally inconvenient configurations: a value of $l_s$ comparable to the width of the holes ensures the removal of as many central points as possible, whereas the small value of $l_s/k_1$ (or even more $l_s/k_1k_2$) enhances the resolution of $P_r$ only where it is necessary, operating inside $P_1$ and $M_1$ (Figure

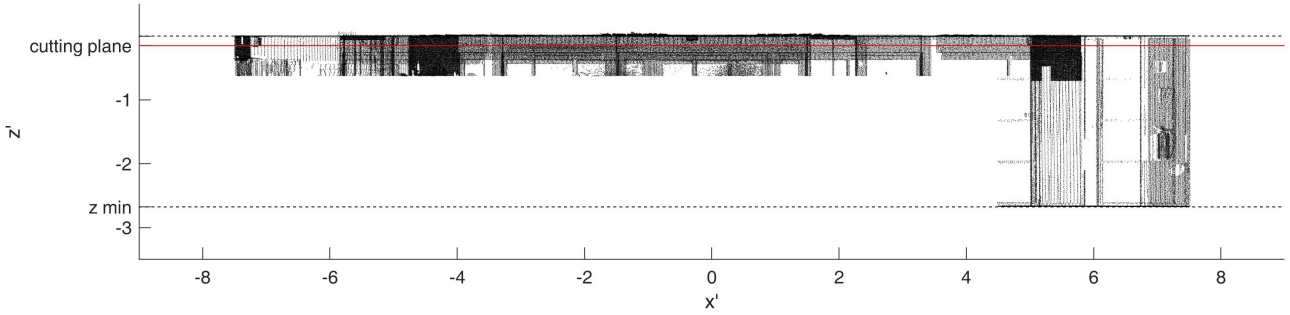



Figure 3: Vertical section of the input point cloud and determination of the room heigth ($h = |z_{min}|$).

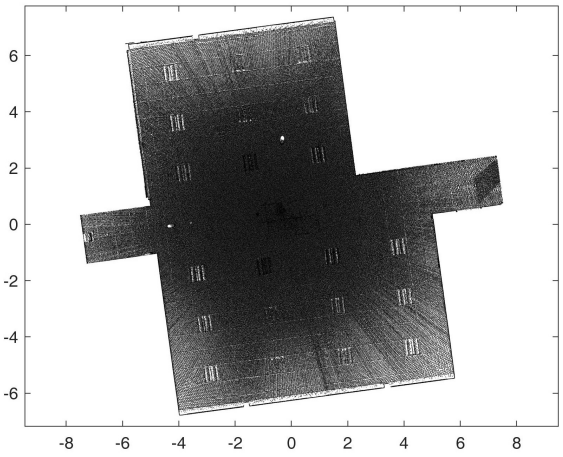
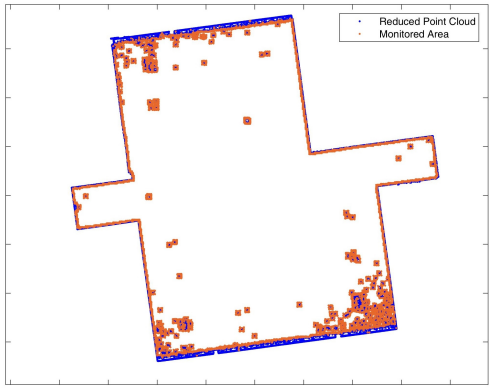
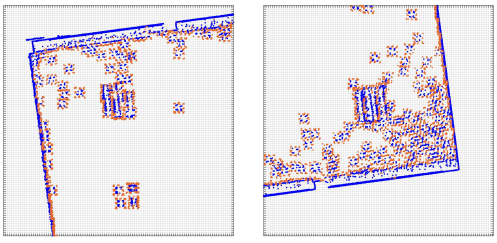


Figure 4: 2D point cloud.




(a)



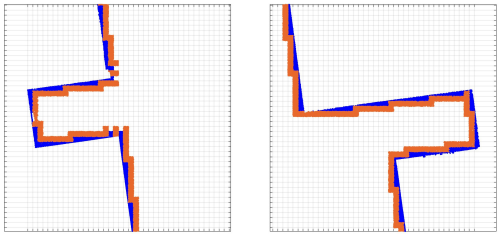
(b)

Figure 5: Reduced point cloud and monitored area for $l_s$ = 3 cm: (a) overview, (b) details.

7). Specifically, a value of $l_s$ comprised between 10 cm and 15 cm (with $k_1 = 4$) was successfully applied in our tests.

Throughout the previous steps, several references to an unspecified boundary zone of the point cloud have been made, but a


(a)



(b)

Figure 6: Reduced point cloud and monitored area for $l_s$ = 15 cm: (a) overview, (b) details.

boundary can only be consistently defined in relation to a subset of the $R_2$ plane rather than the current set of isolated points. Therefore, a Delaunay Triangulation is applied to the reduced point cloud (Figure 8a), imposing a constraint on the maximum edge length, i.e. the Maximum Edge Limit (MEL). A high value for the MEL can cause an alteration in the reconstruction of the final layout because, particularly in concave room, triangular meshes that do not match any real portion of the room are included in the surface from which the boundary will be computed later (Figure 8b). An excessively small MEL produces instead the opposite problem: only isolated portions of the point cloud characterized by small distances between the points are triangulated, and the final layout will not represent the real one at all (Figure 8b).

The correct choice of the MEL parameter depends on the resolution of the sensor employed for the survey: professional laser scanners provide very dense point cloud that can easily be triangulated in full with a MEL of 2 or 3 cm. Instead, the density of a point cloud provided by a low-cost range camera is sensibly lower, so that MELs up to 15 cm may be required. Furthermore, in case of a point cloud originally provided with meshes, as afore mentioned, a statistical analysis can provide a useful advice for the MEL choice: for instance, setting the MEL equal to the maximum edge length of the original triangulation assures a full

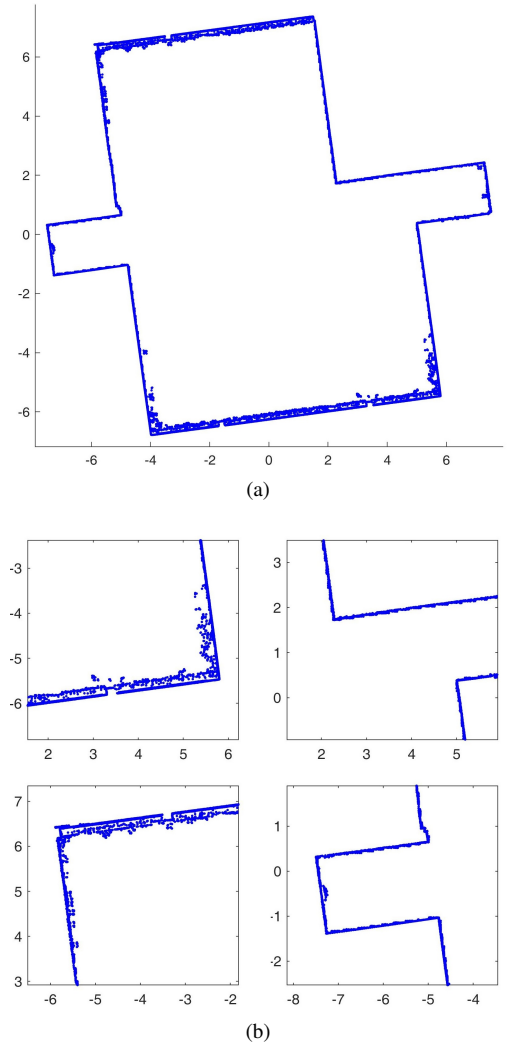


Figure 7: Final reduced point cloud: (a) overview, (b) details.

triangulation of the 2D point cloud, even though a smaller MEL corresponding to the value of the statistical mode could produce an equally optimal result.

Once the final meshes are identified, considering that in a Delaunay Triangulation a segment can only belong to one or two different triangular meshes, it is possible to easily detect the segments of the point cloud boundary because these ones satisfy the first case. These boundary segments form several closed simple polygonal chains (Figure 9), one of which contains all the others and approximates the cloud external border.

However, smaller polygonal chains surround the holes of the point cloud (Figure 10), either the original ones or the biggest one corresponding to the central portion of the cloud removed during the cloud reduction process. The separated polygonal chains are defined as a sequence of boundary segments that share one end-point. The proximity between an eventual hole and the external region, or between two or more holes, can cause the connection of more than two boundary segments through a point $p$ (Figure 11a). This quite frequent situation may lead to ambiguity or make the external polygonal chain merge with another chain. To prevent this issue, all the vertices $p_i$ that are end-point of more than two boundary segments are checked through the following steps:

1. all the segments connected to $p_i$, including the boundary segments (those that we want to identify) and the internal ones, are selected (right panel of Figure 11a);

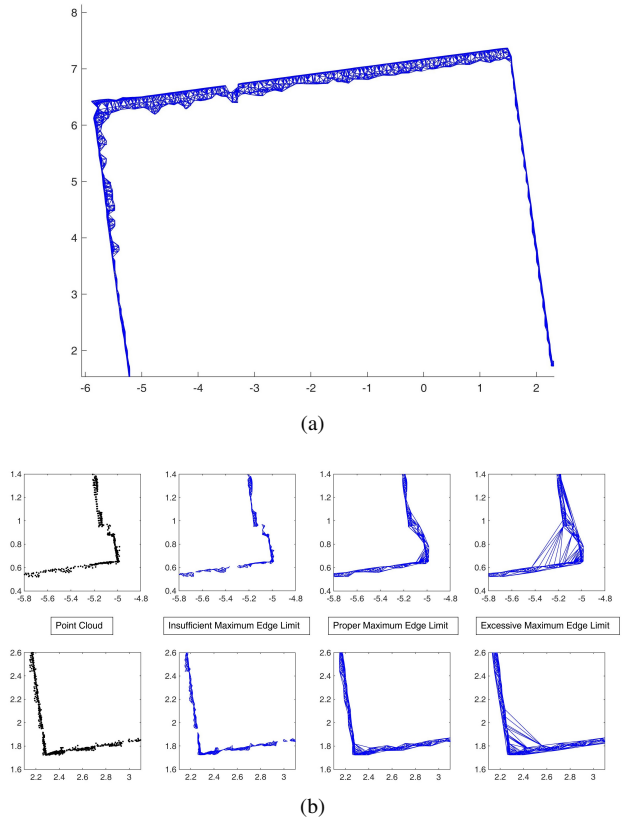



Figure 8: Delaunay Triangulation: (a) detail; (b) influence of the value of the MEL.

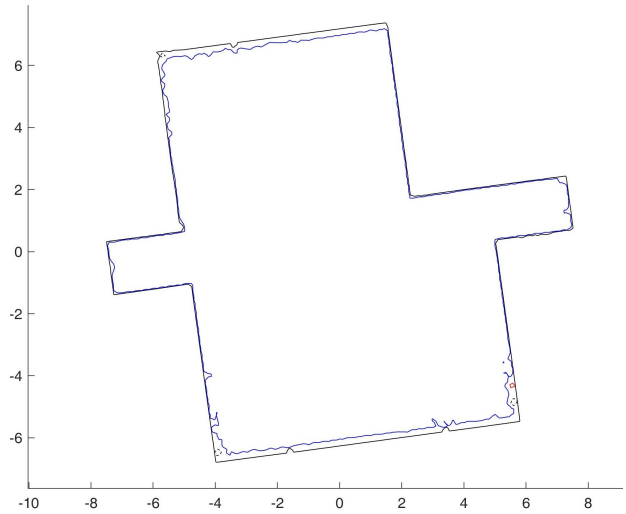


Figure 9: Boundary polygonal chains for the input point cloud.

2. the segments are sorted according to the angle formed with the x axis, counted counterclockwise in the $[0, 2\pi)$ interval;

3. two boundary segments connected to $p_i$ will be in the same polygonal chain only if they are consecutive, or first and last, in the sorted list of the previous step, and if they are not part of the same triangular mesh (Figure 11b);

After this check, it is possible to isolate each polygonal chain from the others, and to select the only one meaningful for the last section of the algorithm: the one that has the same bounding box as the original 2D point cloud (Figure 12).

The last step of the algorithm is aimed to approximate the found

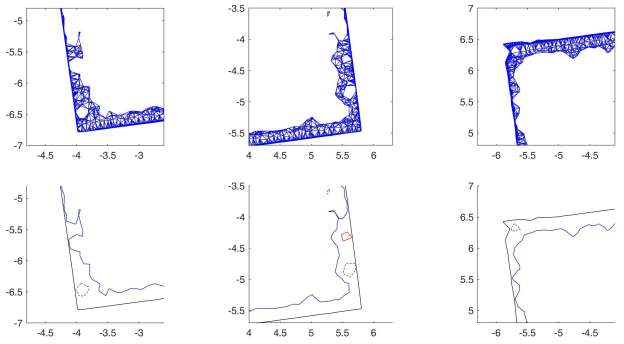
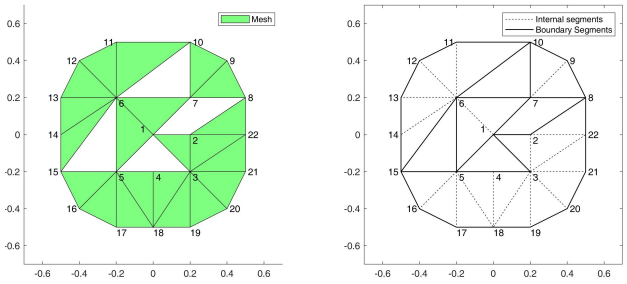
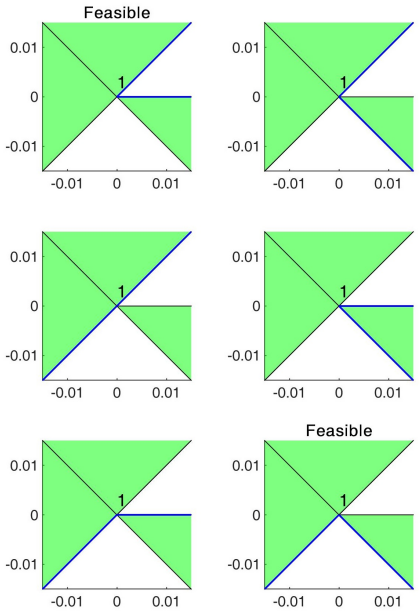


Figure 10: Details of the smaller polygonal chains with relative portions of the Delaunay triangulation.




(a)

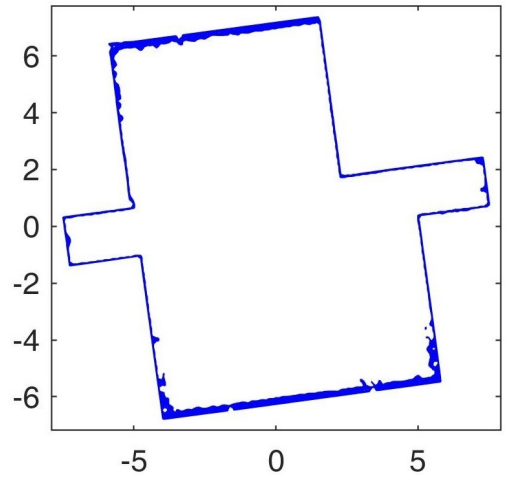



(b)

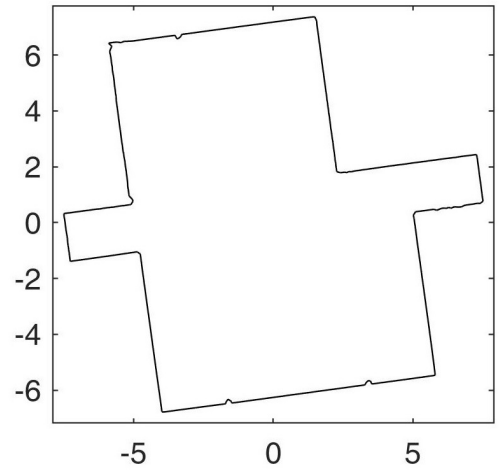
Figure 11: Example of the check procedure on the vertex number 1, which is an end-point for more than two boundary segments.

boundary polygonal chain with a polygon of $n$ sides, where $n$ is equal to the number of sides of the room surveyed. This task requires once again the use of RANSAC, this time employing a linear model. Firstly, RANSAC is recursively applied for $n$ times on the vertices of the polygonal chain, with the outlier set of an iteration serving as the input set for the next one. In each iteration, the RANSAC estimated straight line corresponds to one side of the real room, often the longest available at that point (Figure 13).

If there are two or more sides of the room that are representable as different segments of the same straight line, a check on the inlier set of each RANSAC estimation is needed to avoid that one



(a)



(b)

Figure 12: Selected polygonal chain.

iteration removes the points corresponding to more than one side (Figure 14):

- the maximum distance between a pair of points belonging to the inlier set is computed; one point of this pair is thus selected as a starting point;
- the points of the inlier set are sorted according to the distance from the starting point;
- the distance from every pair of consecutive points in the sorted list is calculated;
- distances exceeding a reasonable maximum distance parameter (for instance 2×MEL) mark a separation within the inlier set: points before and after that distance belong to two (or more) different real side of the room;
- one group is selected as inlier set for the current RANSAC iteration, the other group(s) join the outlier set as input for the next iteration.

This check is useful not only for the above-mentioned case but also for rooms with a concave shape, where the straight line approximating one side can intercept small groups of points located in distant parts of the room. These points are false-inliers and are removed from the inlier set in the same way.

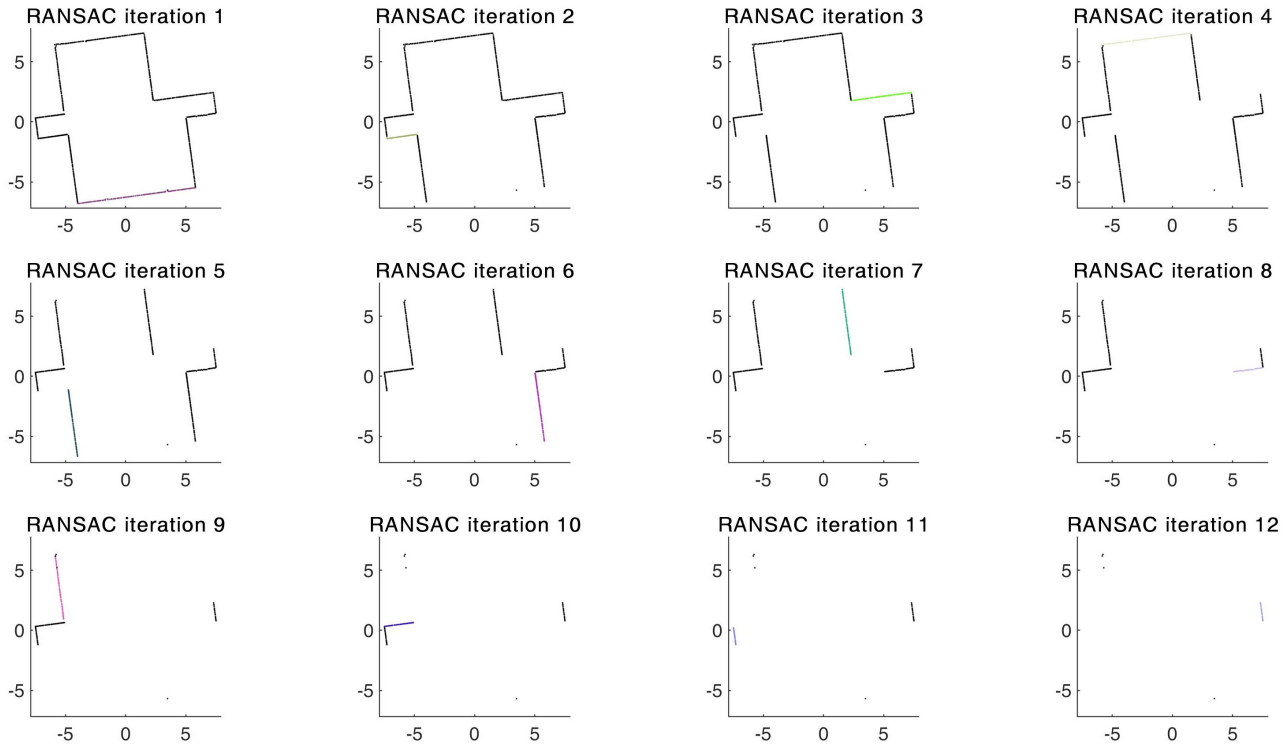



Figure 13: RANSAC iterations for identifying the sides of the room.

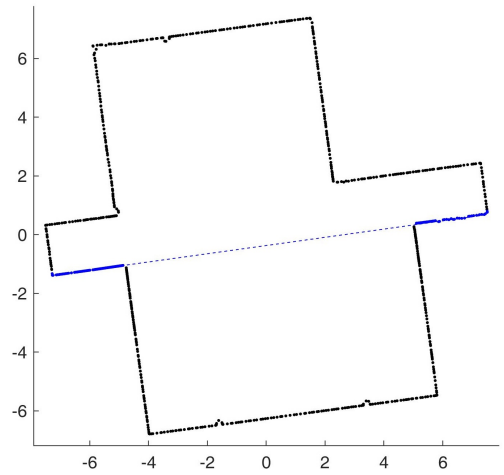


Figure 14: Example of an uncorrect identification of the side.

The choice of the RANSAC threshold parameter depends primarily on the sensor performance in modelling the upper portion of the wall as flat. In the case of a zigzag pattern, typical of low-cost range cameras, a working threshold can be selected only with proper experience. Instead, point clouds collected by laser scanner present a more realistic flat pattern and require smaller thresholds, typically under 5 cm. If the threshold selected is too high, there is the concrete risk that the inlier set includes not only the points belonging to the side of one room, but also few points of the bordering two sides, a circumstance that removes some useful inliers from the estimation of those sides. In order to increase the accuracy, the final RANSAC result is refined by estimating the line parameters only on the inlier set through the Ordinary Least Squares method.

Eventually, the quality of the overall estimation after $n$ iterations is measured by the number of outliers left after the last iteration: for many point clouds it is impossible to bring their number to 0, but the smallest achievable number corresponds in general to the best layout estimation. Hence, the entire estimation ($n$ iterations) described earlier, is repeated indefinitely until the number of final outliers is equal or minor to $n_{estimation}/\alpha$ , where $\alpha$ is a parameter related only to the available computing power. After $\alpha$ estimations, the number of acceptable final outliers increases by 1. Moreover, there are outliers consistently excluded by all RANSAC iterations: they derive from errors in the scanning process, corresponding to objects or physical barriers located in the ceiling-wall border, or to small curved portions of the 3D point cloud (corresponding often to real angles in the real room). To reduce the impact of these points on the algorithm runtime, the points that remain outliers for the totality of the first $\alpha$ estimations are removed from the input set of the subsequent estimations; same goes, if necessary, for steady outliers occurring in the second or third section of $\alpha$ estimation, although this is a rarer circumstance.

Once the final set of inliers is found, a raw layout is available (Figure 15): it consists of one line equation and two endpoints expressing the position of the inlier set, for each side of the room.

Finally, the actual layout is produced extending the line segments along their lines, two by two, up to the point of their intersection (Figure 16).

## 3 RESULTS

As mentioned before, the algorithm developed was tested on the 3D point cloud of 600.000 vertices selected from the ISPRS dataset on indoor modelling. The obtained layout is reported in Figure 17: the results are very promising, the algorithm is able to model effectively the 3D shape of the investigated room, at least from a visual inspection. Furthermore, a height of 2.67 m was estimated for the input point cloud.

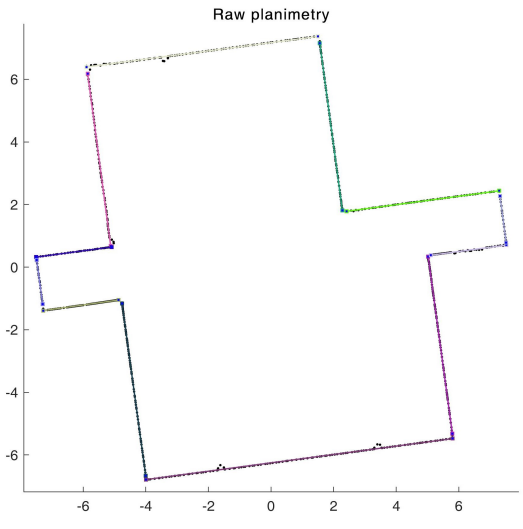



Figure 15: Raw layout.

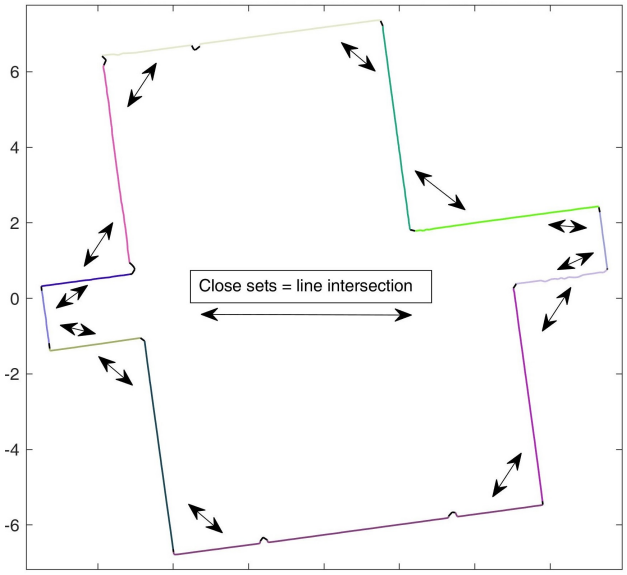



Figure 16: Inlier locations along the border of the polygonal chain. In black the outliers.

As the computational efficiency, the algorithm average runtime on the input point cloud is, at the moment, five minutes on a Desktop PC with an IntelCore i7 3 GHz CPU and 16 GB of RAM. Both the runtime and the system requirements can be considerably narrowed with informatic expertise: the parallelization of the code and the switch to more efficient and fast languages could optimize the CPU use, while a careful selection of the Data Structure could avoid bottlenecks and an excessive usage of RAM resources. Furthermore, the portability of the software is threatened only by the amount of data that have be processed in the 3D section of the algorithm, even though it is possible to consider a reduction of the cloud density, based for instance on a required minimum distance between every couple of points.

## 4 CONCLUSIONS AND FUTURE WORK

In this paper, a new algorithm for the automatic extraction of the layout and the height of a small environment was presented. In particular, the algorithm was tested on a point cloud of 600000 vertices, selected from the ISPRS benchmark on indoor modelling (Khoshelham et al., 2017).

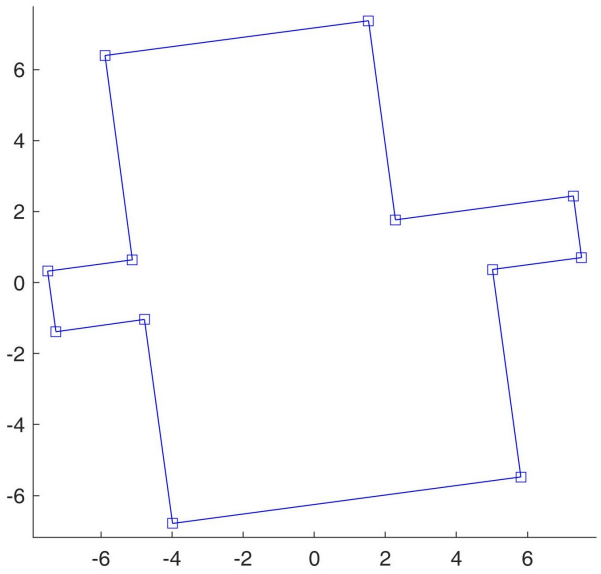


Figure 17: Final layout reconstructed.

The preliminary results are encouraging: the 3D shape (layout and height) of the investigated room is effectively reconstructed. However it is still necessary to deepen the analysis, by testing the algorithm on more and different environments, performing also a quantitative accuracy assessment, using for instance the evaluation criteria described in (Khoshelham et al., 2018).

Furthermore, at this stage, only room with polygonal layout and flat surfaces for walls and ceiling are taken into consideration, but thanks to the adaptability of RANSAC to different models in both its 2D and 3D variants, curve-shaped room together with room characterized by slanted walls or ceiling, could soon become feasible input. Finally, a possible future development is to release the implementation of the algorithm as a Free and Open Source Software (FOSS).

## REFERENCES

Capocchiano, F., Ravanelli, R. and Crespi, M., 2017. A tool for crowdsourced building information modeling through low-cost range camera: Preliminary demonstration and potential. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLII-2/W8, pp. 75–81. doi: 10.5194/isprs-archives-XLII-2-W8-75-201.

Fischler, M. A. and Bolles, R. C., 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.

Khoshelham, K., Díaz Vilariño, L., Peter, M., Kang, Z. and Acharya, D., 2017. The ISPRS benchmark on indoor modelling. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLII-2/W7, pp. 367–372. doi: 10.5194/isprs-archives-XLII-2-W7-367-2017.

Khoshelham, K., Tran, H., Díaz-Vilariño, L., Peter, M., Kang, Z. and Acharya, D., 2018. An evaluation framework for benchmarking indoor modelling methods. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLII-4, pp. 297–302. doi: 10.5194/isprs-archives-XLII-4-297-2018.

Ravanelli, R., Nascetti, A. and Crespi, M., 2015. An Open Source RANSAC based Plug-In For Building Roof Extraction From LiDAR Point Clouds. In: The 35th EARSeL Symposium European Remote Sensing: Progress, Challenges and Opportunities - Abstract proceedings.

Sansò, F., 1973. An exact solution of the roto-translation problem. Photogrammetria 29, pp. 203–216. doi: 10.1016/0031-8663(73)90002-1.

Xiong, X., Adan, A., Akinci, B. and Huber, D., 2013. Automatic creation of semantically rich 3D building models from laser scanner data. Automation in Construction 31, pp. 325–337. doi: 10.1016/j.autcon.2012.10.006.