

Article

# Indoor Building Reconstruction from Occluded Point Clouds Using Graph-Cut and Ray-Tracing

Mattia Previtali <sup>1</sup>, Lucía Díaz-Vilariño <sup>2,\*</sup>  and Marco Scaioni <sup>1</sup> 

<sup>1</sup> Politecnico di Milano, Department of Architecture, Built Environment and Construction Engineering, Via Ponzio 31, 20133 Milano, Italy; mattia.previtali@polimi.it (M.P.); marco.scaioni@polimi.it (M.S.)

<sup>2</sup> Department of Natural Resources and Environmental Engineering, University of Vigo, Campus Lagoas-Marcosende, CP 36310 Vigo, Spain

\* Correspondence: lucia@uvigo.es; Tel.: +34-986-813-499

Received: 4 July 2018; Accepted: 16 August 2018; Published: 1 September 2018



**Abstract:** Despite the increasing demand of updated and detailed indoor models, indoor reconstruction from point clouds is still in an early stage in comparison with the reconstruction of outdoor scenes. Specific challenges are related to the complex building layouts and the high presence of elements such as pieces of furniture causing clutter and occlusions. This work proposes an automatic method for modelling Manhattan-World indoors acquired with a mobile laser scanner in the presence of highly occluded walls. The core of the methodology is the transformation of indoor reconstruction into a labelling problem of structural cells in a 2D floor plan. Assuming the prevalence of orthogonal intersections between walls, indoor completion is formulated as an energy minimization problem using graph cuts. Doors and windows are detected from occlusions by implementing a ray-tracing algorithm. The methodology is tested in a real case study. Except for one window partially covered by a curtain, all building elements were successfully reconstructed.

**Keywords:** indoor modelling; BIM; energy minimization; graph cut; regularization; terrestrial laser scanning

## 1. Introduction

Up-to-date 3D indoor models are being increasingly requested for the management of existing buildings. The modelling of as-built constructions can range from mesh and computer-aided design (CAD) models to building information models (BIM). While the two first representations are enough for visualization purposes, the reconstruction of semantically-rich representations such as BIM is required for a variety of applications in which, apart from the detailed geometry, semantics and topology play an important role. These applications are mainly routing and navigation [1], crisis management [2], energy efficiency analysis [3], and structural health monitoring, inter alia.

In spite of the increasing demand for updated and detailed indoor models, the reconstruction of indoor scenes continues to be in an early stage in comparison with outdoor reconstruction [4]. Dense point clouds, provided by reality-capture technologies such as laser scanner [5] and photogrammetry [6], are typically processed manually or semi-automatically to create 3D models of buildings. Even if the automatic reconstruction of buildings outdoors shares many properties (and problems) with the issues associated with indoor modeling, there are some specific challenges due to complex building layouts and the high presence of objects like furniture and wall-hangings causing clutter and occlusions [7].

The ongoing advances in the reduction of size and weight of terrestrial laser scanning sensors, together with improvements in indoor positioning techniques, have led to the development of productive Indoor Mobile Mapping Systems (IMMS) [5]. These systems may be cart-based [8],

backpack-based [9] or handheld-based [10] depending on the platform in which they are carried. The dynamic nature of the acquisition process has as a consequence a higher availability of 3D point cloud data from indoors, especially in terms of data completeness. In addition, not only the point cloud but also the trajectory followed by the system during the acquisition is usually available. This is because most of these systems are based on simultaneous localization and mapping (SLAM), which is a technique consisting of the construction of an incremental map of the unknown environment and the simultaneous localization within it [11].

The goal of this research was to develop an automatic method for modelling Manhattan-World (MW) indoor scenes on the basis of laser scanner data. An MW assumes that most man-made structures may be approximated by planar surfaces that are parallel to one of the three principal planes of a common orthogonal coordinate system. The presented methodology specifically copes with the presence of significant amount of clutter and occlusions that may cause a significant lack of data. The main assumption is the consideration of planar surfaces bounding the rooms. The large majority of existing residential buildings in modern cities are in line with this assumption. Doors and windows are also reconstructed. A ray-tracing algorithm was used to identify those regions that are occluded from every viewpoint and to distinguish these regions from openings in the surface.

The paper is organized as follows. Section 2 reviews the state-of-the-art techniques proposed in the recent literature for indoor modelling. Section 3 describes the methodology to be adopted for indoor reconstruction. Section 4 explains the conducted experiments and respective results and Section 5 is devoted to the conclusion of this work.

## 2. State-Of-The-Art

Most of the applications requiring indoor models need semantically-rich representations including not only geometry, but also semantics and topology. Although BIM is the traditional domain encoding all the relevant information of individual buildings, Geographic Information Systems (GIS) have become increasingly detailed and have begun to model 3D cities including indoors. In spite of the fact that GIS and BIM domains overlap in terms of building modelling, each domain still has its own focus and characteristics. BIM considers very detailed and semantically rich information on all the physical elements comprising individual buildings as they are designed or built. On the other hand, 3D GIS representations mainly supported by the CityGML data model are more focused on describing the information about the environment that is captured at different points in time. In this case, representations of indoor scenes are less detailed but included into the broad context of a city. Apart from CityGML, the Open Geospatial Consortium (OGC) published IndoorGML, an application schema of the Geography Markup Language (GML), focusing also on semantics, geometry, and topology of indoor environments. Nevertheless, this scheme does not extensively cover geometric and semantic properties to prevent overlap with other indoor building modelling standards such Industry Foundation Classes (IFC) data model and CityGML.

Several methods have been proposed in the recent literature for generation of indoor models. Their performance can hardly be compared since it mostly depends on the level of clutter and occlusions of the input point cloud and on the geometric, topological, and semantic detail of the output model. From this requirement, the ISPRS Benchmark on Indoor Modelling proposes the creation of a common framework for the evaluation and comparison of indoor modelling methods but it is still under development [12].

Recent methods dealing with indoor modelling can be classified into three categories according to the primitives they are reconstructing: linear-primitive, planar-primitive, and volumetric-primitive based [7].

In the first category, indoor modelling is based on floor plan extraction followed by extrusion, assuming walls as planar and vertical surfaces. These methods are generally performed on isolated floor levels and in the absence of clutter. Oesau et al. [13] develop a methodology for modelling floor plans by applying cell decomposition after line fitting followed by a graph cut optimization. Ref [14]

reconstructing buildings indoors from point clouds already distributed into separate rooms. In this case, the labelling step is also solved as an energy minimization problem. Both works apply to MW and non-MW structures. However, they deal neither with occluded wall surfaces nor with door and window reconstruction.

The planar-primitive approach consists of submitting point clouds to a planar primitive detection followed by a classification. Sanchez et al. [15] use Random Sample Consensus (RANSAC) to plane fitting and calculate their extents by using alpha shapes. Similarly a plane-sweep approach is used in [16,17] to find planar regions. However, even if these algorithms work well for extracting planar patches from the laser data, they do not consider the complete occlusion problem. These approaches are generally using 'contextual-based' reasoning to distinguish between building elements prior to plane fitting and intersection. Such approaches generate semantic labels of geometric primitives, and test the validities of these labels with a spatial relationship knowledge base [18]. The rule-coding process is generally quite complex and missing data may cause an incorrect application of rules.

The volumetric-primitive detection generally imposes a stronger regularity and the developed methods are so far restricted to MW structures. Furukawa et al. [19] introduced an inverse constructive solid geometry algorithm based on detecting wall segments in 2D sections which are combined to generate cuboids. Khoshelham and Díaz-Vilariño [20] developed a grammar-based methodology to reconstruct MW indoor spaces by iteratively placing, connecting, and merging cuboids. This methodology was further extended to extract the topological relations between indoor spaces for indoor navigation [21]. These approaches are productive in large-building reconstruction but they have been tested in the absence of clutter.

Apart from building bounding modelling, doors and windows are also permanent structures in indoor scenes and several methods have been proposed in recent years. A voxel-based labelling approach based on a visibility analysis [22] was proposed by [23,24]. The Generalized Hough Transform is used on wall orthoimages generated from colored point clouds to detect closed doors [25]. The methodology was further extended to classify door candidates into open and closed doors by studying the point-to-plane distribution of points close to the door candidate [26]. Another approach based on orthoimages is the one presented by [27], in which doors are detected by finding horizontal and vertical lines. Trajectories are also used to detect doors crossed during data acquisition with an IMMS [7]. Assuming doors as typically lower in height than the walls they are contained, doors are extracted from searching peaks in a vertical profile of the point cloud extracted along the trajectory. Another approach based on a ray-tracing algorithm to detect openings from occluded areas in a voxelized point cloud was implemented by [28].

### 3. Indoor Reconstruction Methodology

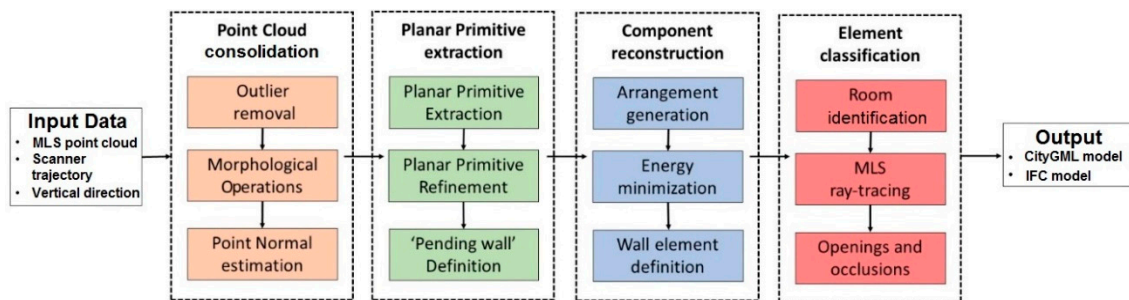
#### 3.1. Overview

As previously anticipated, in this paper we present a technique for automatic indoor modelling (AIR) of buildings starting from point cloud data acquired with a mobile laser scanner (MLS). One of the key-points of the methodology is the capability to cope with a large amount of clutter and missing data.

The developed methodology takes as input a point cloud from indoor MLS ( $x$ - $y$ - $z$  coordinates), its trajectory, and a defined vertical direction (that generally coincides with the Z axis). From a formal point of view, the point cloud can be represented as a set of points  $P = \{p_i \mid i \in \{1, \dots, N\}\}$  in  $R^3$  while the continuous trajectory of the mobile laser scanner can be discretized into a set of points  $TR = \{tr_k \mid k \in \{1, \dots, M\}\}$  representing the scanner position during the acquisition process.

These prerequisites can be easily obtained in practice. Indeed, commercial and scientific MLS generally provide both point clouds of the crossed rooms and the scanner trajectory as an output of the acquisition step. In addition, the point clouds are provided with the  $z$  axis aligned to the vertical direction.

The presented methodology uses the previously listed inputs to reconstruct the indoor scene through a planar primitive detection and an indoor component reconstruction process. The adopted workflow is summarized in Figure 1.



**Figure 1.** Workflow of the developed methods for automatic indoor modeling (AIR) of building rooms. MLS = mobile laser scanner

In particular, since point clouds recorded with MLS are often corrupted with noise and outliers a first smoothing is carried out by using some statistical analysis and morphological operators (removal of isolated points). The consolidation of the raw point cloud reduces noise and increases the accuracy of the subsequent reconstruction steps. The normal for each point is then estimated using local plane fitting.

The core of the presented methodology is the transformation of the indoor reconstruction into the labelling problem of structural cells in the 2D floor plan. More precisely, the first step is the detection and estimation of the surfaces to be modelled (i.e., walls, ceilings, and floors). In particular, the extraction of the potential indoor primitives is detected by using a modified RANSAC implementation for extraction of planar surfaces. The extracted primitives are then refined to reduce under- and over-segmentation issues. The refined primitives are then considered as the input for the subsequent indoor component reconstruction.

However, due to occlusions and clutters, some walls may be missing in the data set. For this reason, an automated procedure is implemented to look for pending elements in a plausible way. To achieve this, the developed algorithm incorporates some architectural priors on indoor scenes, notably the prevalence of orthogonal intersection between walls—the Legolan building assumption [29]. In particular, the indoor reconstruction is formulated as an energy minimization problem using graph cuts to finalize the creation of walls.

Once the wall elements are defined, the remaining steps operate on each surface individually. In this phase, each planar surface is analyzed to identify and to distinguish between occluded regions and openings by using a ray-tracing algorithm [30]. This enables the identification and labeling of doors and windows, so that finally the occluded regions may be reconstructed in a realistic way.

The output of the methodology consists of a set of labeled planar patches (walls, floor, and ceiling), adjacency maps indicating connection between planes, and a set of openings detected within each planar patch. These patches are intersected one with another to form a simple surface-based model of the room. The transformation of a surface-based model into a volumetric based one is carried out by means of the commercial software Rhinoceros<sup>®</sup>. The geometric nodes of the room, along with any semantic information may be combined together to derive a semantically enriched model in CityGML and/or IFC format [31]. Indeed, both formats provide the description of building features not only as a simple geometry but also as the class of the feature it belongs to (i.e., its semantic). However, the semantic classification of the two formats is quite different both in terms of Level of Information (LoI) and in terms of class definition. In such a way, the reconstructed 3D model can be used to start the population of the BIM with further information (e.g., materials, etc.).

### 3.2. Point Cloud Consolidation

Raw indoor MLS point clouds often contain noise, and outliers especially near the openings (e.g., opened doors) and transparent regions (e.g., windows). This is mainly due to the fact that the laser scanner beam can pass through such regions causing sparse 3D points. In addition, MLS data presents generally much more noise than traditional static Terrestrial Laser Scanning (TLS) data that may arise in the wrong segmentation of the point cloud causing the generation of artifacts. To reduce the interference of subsequent planar primitive extraction the point cloud is first preprocessed with consolidated through noise smoothing and outlier removal. The noise smoothing is carried out by a statistical analysis of the point cloud. Since the raw data may contain outliers due to random measurement errors, for each point  $p_i$  the average distance ( $d_i$ ) to its nearest  $k$  neighbors is computed. Assuming that the result of average distances has a Gauss distribution, points falling outside a confidence interval  $[\mu_{di} - 2\delta_{di}, \mu_{di} + 2\delta_{di}]$  are discarded. Parameters  $\mu_{di}$  and  $\delta_{di}$ , represent the mean value and the standard deviation of the distances, respectively. A second removal filter is applied to remove isolated points. This filter uses both mathematical morphological opening operations and connected component analysis. In particular, starting from the point cloud, a volumetric voxel representation is created with a grid resolution  $V_R$ . Then, the occupancy of each voxel is evaluated. In particular, a binary representation of the space is provided:

$$B(x, y) = \begin{cases} 0, & n_c = 0 \\ 1, & n_c > 0 \end{cases} \quad (1)$$

where  $n_c$  is the number of points in each voxel. Morphological opening operations are then employed to filter out isolated points. Indeed, a connected component analysis is used to detect the connected component in the voxel space. Making again the assumption that the size of the different connected component features a Gaussian distribution, the components whose size falls outside the interval  $[\mu_{si} - 2\delta_{si}, \mu_{si} + 2\delta_{si}]$  are filtered out as isolated elements, while points belonging to those voxels are discarded. Parameters  $\mu_{si}$  and  $\delta_{si}$ , represent the mean value and the standard deviation of the size of connected components, respectively. This voxel space is also used for the ray-tracing algorithm (see also Section 2).

Finally, starting from the point cloud after preliminary filtering, the local normal is estimated for each point. This task is operated by using the method proposed in [32]. For each point a plan is locally fitted considering a certain number of neighboring points, the number of considered points is iteratively increased until the estimated normal and the eigenvalues of the weighted covariance matrix  $C(N_k(p))$  reach stabilization:

$$C(N_k(p)) = \sum_{j \in N_k(p)} (p_j - \mu_k)(p_j - \mu_k)^T \phi(\|p_j - p\|/h_k) \quad (2)$$

where  $\mu_k$  is the weighted average of all points in  $N_k(p)$ ,  $h_k$  is the radius of the smallest sphere containing  $N_k(p)$  centered at  $p$ , and  $\phi$  is a positive monotonously decreasing weighting function.

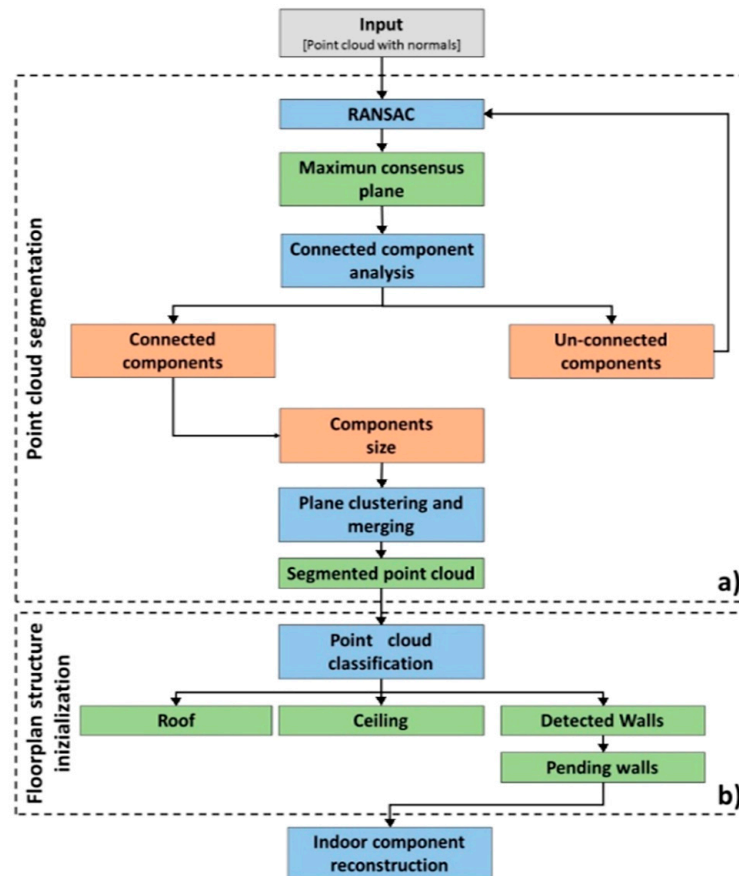
### 3.3. Planar Primitive Extraction

Once the point cloud is consolidated and for each point  $p_i$  its local normal  $n_i$  is computed, a planar primitive detection is carried out to identify potential planar room surfaces (walls, roofs, and ceilings). This stage is accomplished by using the segmentation strategy described in [33] (see also Figure 2, part a). In particular, the planar primitive detection is carried out by using a hybrid technique combining RANSAC algorithm [34] and region growing. In this implementation, major attention is paid to the reduction of so called 'bad segmentation' problems. Indeed, spurious results may be due to the fact that points constituting the maximum consensus to RANSAC planes are derived from different



objects. To reduce the effect of outliers and fine structures such as bumps and craters, the consensus to the RANSAC score function  $F$  takes into account the following aspects:

- the number of points that fall within the  $\epsilon$ -band around the plane; and
- to ensure that the points inside the band roughly follow the direction of a given plane, only those points inside the band whose normal do not deviate from the normal of the plane more than the defined angle  $\alpha$  are considered as inliers for the guessed plane.



**Figure 2.** Workflow for planar primitive extraction: the two main steps are the segmentation of the point cloud using Random Sample Consensus (RANSAC) (a) and the semantic classification of detect surfaces into walls, floor, and ceiling (b).

More formally, given a candidate shape  $C$  whose fidelity is to be evaluated,  $F$  is defined as follows:

$$F(C) = |P_\psi| \tag{3}$$

i.e.,  $F(C)$  counts the number of points in  $P_\psi$ , which is defined as:

$$P_\psi = \{p \mid p \in P \wedge |d(\psi,p)| < \epsilon \wedge \arccos [n(p) \cdot n(\psi,p)] < \alpha\} \tag{4}$$

where  $d(\psi, p)$  is the signed distance of point  $p$  to the plane  $\psi$ ,  $n(p)$  is the normal in  $p$  and  $n(\psi,p)$  is the normal of  $\psi$  in  $p$ 's projection on  $\psi$ . In particular, the signed distance function for a plane is given by:

$$d(x) = \langle n, x - p \rangle = \langle n, x \rangle - \langle n, p \rangle \tag{5}$$

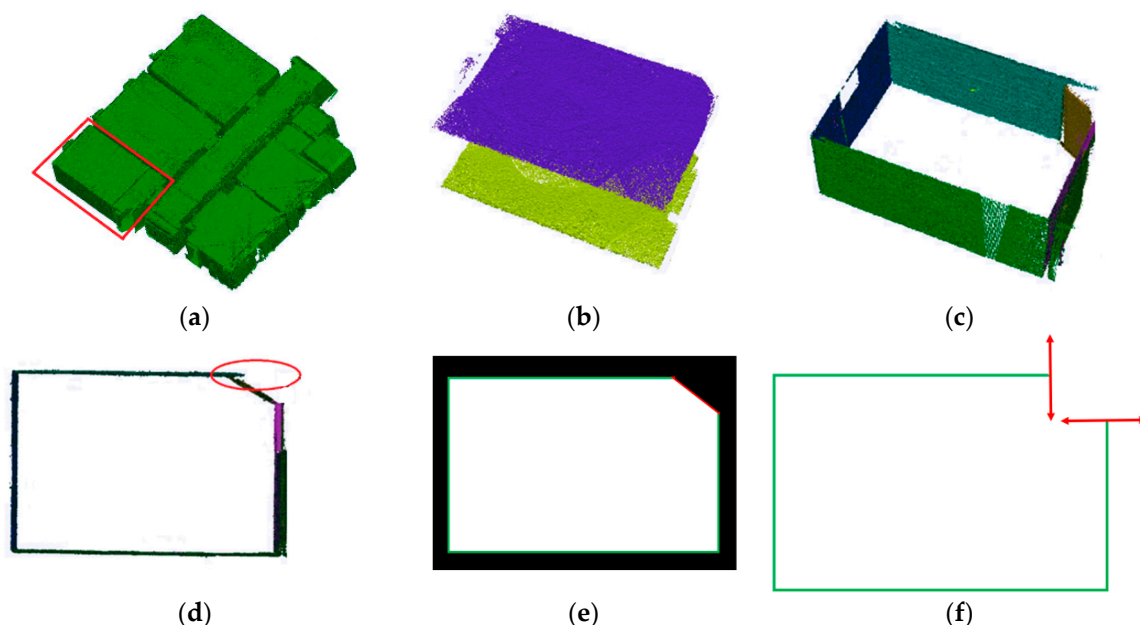
where  $n$ ,  $|n| = 1$  is the normal to the plane and  $p$  is an arbitrary point in the plane. The intuitive threshold value  $\epsilon$  for the Euclidean distance between a point and an estimated plane can be easily found

by the user according to the instrumental noise and minimum point density for the acquired point clouds. All plane primitives whose supporting points are less than a threshold  $n_r$  are directly discarded.

The developed strategy tries to minimize spurious objects by refining the detected elements including topology information into the process. Indeed, points belonging to the same object should be sufficiently close while groups of points belonging to different objects should have a gap area. For this reason, a topology measure is introduced in the segmentation process by identifying objects supported by a connected component point data set. Similarly, once all objects are detected, over-segmented parts are combined together considering the topology properties of the extracted planes (i.e., similarity of normal vectors, perpendicular distance between planes and planes' intersection). Clustering of detected elements is performed by using mean shift clustering [35].

Once the point cloud segments have been detected, a first semantic classification is operated to detect surfaces which constitute the 'room box' including walls, floor, and ceiling, see also Figure 2, part b. Indeed, detected segments may also include planar pieces of furniture (e.g., tables) that have to be excluded from the indoor modelling. In Figure 3 some steps related to the contour extraction for one room of the dataset used in the paper (see also Section 3) are shown. In particular, roof and ceiling surfaces can be easily detected looking for horizontal elements. Indeed, the floor can be designed as the horizontal plane having the lower height (Figure 3a,b). Conversely the ceiling is detected as the horizontal plane located at the highest level. In this way, the distance from floor to ceiling can be also worked out.

To determine the floorplan, first the walls of the room need to be detected (Figure 3c). This problem can be difficult due to possible clutter and occlusions, resulting in the fact that walls may have not been scanned and thus are pending in the point cloud. For this reason a proper completion is necessary in order to reconstruct these walls in a plausible way (Figure 3d). A first rough floorplan can be obtained by projecting the points belonging to the ceiling onto a horizontal plane. Indeed, the acquisition of the ceiling surface, due to its location, is generally less influenced by clutter and occlusions than other surfaces of the room.

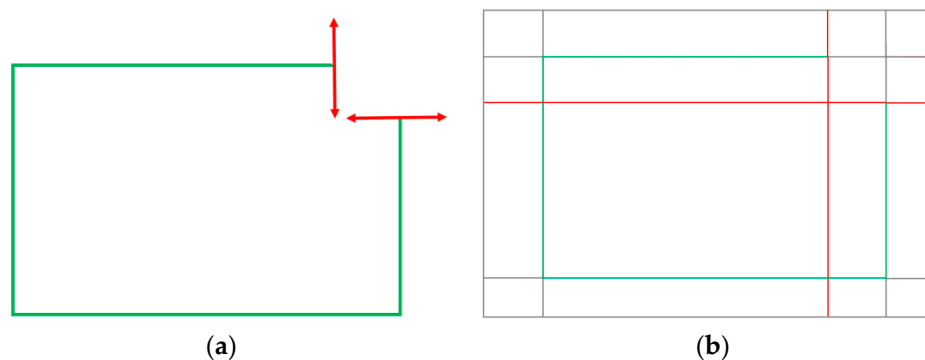


**Figure 3.** Detection of room surface: (a) localization of the room addressed in this example; (b) planar elements classified as ceiling (purple) and floor (sand); (c) possible wall surfaces; (d) a wall portion is missing (red circle) due to occlusions, (e) occupancy map of the ceiling with real walls (green segments) and spurious boundaries (red segments) for Room 1, (f) detected primitives (green) and guessed 'pending' walls.

Usually, indoor environments with a standard usage feature most of the floor surface free for walking and moving around. The horizontal plane is discretized into cells of size  $\beta_1 \times \beta_1$  where  $\beta_1$  is set equal to the mean sampling resolution in the point cloud. Then an occupancy map is generated where one pixel represent elements where MLS data are available, and zero pixels are grid elements with no data. Starting from this binary rate map it is possible to derive pixels representing the boundary of ‘occupied’ cells which represent a first rough floorplan. Due to possible occlusions the obtained planes may contain some spurious boundaries, i.e., the ones not associated to a wall (Figure 3e). To validate the obtained boundaries, a check is done against the segmentation results. In particular, only vertical segments falling inside the cells labelled as boundary are considered as real wall surfaces. In indoor modeling applications, a single pending small wall may jeopardize the entire reconstruction of the floor plan. In the developed strategy, such gaps are filled by incorporating additional, unseen ‘pending’ walls [36]. In the indoor environment it is possible to observe that walls generally intersect orthogonally. For this reason ‘pending’ walls are guessed as orthogonal to the already detected and are added from the boundary of the recognized walls (Figure 3f).

#### 3.4. Indoor Component Reconstruction

Starting from the detected and the guessed ‘pending’ wall derived from the planar primitive extraction phase (Figure 4a), the indoor building model is derived by applying a decomposition and labelling procedure based on cell complex labelling. To this end, the floorplan is decomposed into adjacent cells creating a 2D arrangement [37] which is set up by using both ‘detected’ and ‘pending’ walls. The cells are then labelled as ‘indoor’ and ‘outdoor’ by solving an energy minimization problem by using graph cut optimization [38]. The cell decomposition technique has been widely used in building ground plane generalization [39]. In particular, starting from the detected and ‘pending’ walls identified in the previous step the ground plan is partitioned into a set of cells generating a partitioning of the original space domain into convex polygonal cells (Figure 4b). After the cell decomposition, we obtain a 2D arrangement composed of a set of cells  $T = \{T_k \mid K = 1, \dots, k\}$ , which are labelled as ‘occupied’ (within the room) and ‘empty’ (outside the room) cells.



**Figure 4.** Example of cell complex construction: (a) detected and ‘pending’ walls and (b) induced cell complex.

Once having derived the cell complex, the floor plan reconstruction problem may be formulated as an optimal binary labelling of cells in the complex. Each cell is labeled as ‘empty’ or ‘occupied’ and the floor plan can be extracted as the union of all facets separating an ‘occupied’ cell from an ‘empty’ one. In this way an intersection-free boundary may be obtained. Graph cut is an optimization method for solving this type of binary labelling problem through a minimum cost cut on a graph [40]. An energy minimization score function is defined and the minimization problem can be handled using global graph cut optimization.

A unidirectional graph  $G = (V, E)$  is defined as a set of nodes and a set of unidirectional edges to encode the set of cells  $T$  and the label set  $L = \{L_{empty} \text{ and } L_{occupied}\}$ . The set of decomposition cells  $T =$



$\{T_k | K = 1, \dots, k\}$  can be assigned to a set of vertices  $V = \{V_k | K = 1, \dots, k\}$  connected with weighted edges. Each edge in the graph is assigned with a non-negative weight  $W_e$ . The set of edges  $E$  is composed of two parts:  $n$ -links (neighborhood links) and  $t$ -links (terminal links). Therefore, the set of edges may be expressed as  $E = N \cup U_v \in V \{\{v,S\},\{v,T\}\}$  where  $N$  is the set of neighborhood links, while  $S$  and  $T$  represent the “source” and the “sink” model expressing foreground and background in the graph cut respectively. In particular, the source node is labelled as the empty space  $L_{empty}$  while the sink is the occupied space  $L_{occupied}$ . In other words, the vertices  $V$  are the cells of the polygonal cell complex and the edges  $E$  link adjacent cells, i.e., they correspond to the facets of the complex augmented with two additional seeds, a source  $s$  and a sink  $t$ , with edges from  $s$  to each cell and from each cell to  $t$ . All edges have non-negative weights  $W$ . A  $s - t$  cut  $(S,T)$  is a partition of  $V$  into two disjoint sets  $S$  and  $T$  such that  $s \in S$  and  $t \in T$ . The cost of an  $s - t$  cut is the sum of the weights of the edges from  $S$  to  $T$ . An efficient algorithm with low-polynomial complexity exists to find the  $s - t$  with minimal cost, allowing a global minimization of the energy. The graph partitioning  $(S,T)$  corresponds to a binary labelling of cells (Figure 5), where cells in  $S$  and  $T$  are respectively empty and occupied, and the cost of the cut corresponds to the energy of the associated surface. Weights of those edges joining the source or the sink penalize the associated cells, while weights of those edges between two cells penalize the associated facets. In particular, the energy function to be optimized is the following one:

$$E = \lambda \sum_{i \in V} D_i(l_i) + (1 - \lambda) \sum_{(i,j) \in N} S_{i,j}(l_i, l_j) \tag{6}$$

where  $\sum_{i \in V} D_i(l_i)$  represents data term, i.e., the penalty of assigning every vertex to the  $t$ -link, while  $\sum_{(i,j) \in N} S_{i,j}(l_i, l_j)$  is the smoothness term representing the cost of  $n$ -links between the adjacent vertices.  $S_{i,j}$  is the pairwise cost for adjacent cells to ensure a smooth segmentation. Finally,  $\lambda$  is a weight in order to balance the data term and the smoothness term.

### 3.4.1. Data Term

The data term encloses the weights between the cells and the terminal nodes. Starting from the available data, some cells can be directly categorized as occupied (Figure 5a). In particular, all cells occupied by points belonging to the ceiling can be directly assigned to set  $T$ . In a similar way, cells bordering an occupied cell and separated from it by a detected wall segment are set as empty. For this reason, weights of edges joining the sink to cells labelled as occupied are set to infinite and, in a similar way, edges joining the source to empty cells are set to infinite. The weights of the remaining edges between cells and terminal nodes are presented by using a method based on the ray casting method inspired by that found in [13,41]. This method essentially creates virtual rays connecting the scanner positions (which in the case of an MLS changes during the time) and a cell. If the rays intersect an edge detected as “real” in the previous step the tracing is stopped. On the other hand, in the case the ray traverses only “pending” walls the ray is cast. The rationale of this method is that the cells belonging to the interior space are more likely to intersect the rays than the exterior cells. In other words, the outer cells will have relatively low weights and the interior cells will have relatively large weights. More specifically, summarizing the previously listed cases we have:

If the cell is classified as occupied starting from ceiling occupancy:

$$D_i(l_i) = \begin{cases} \infty & l_i \in ext \\ 0 & l_i \in int \end{cases} \tag{7}$$

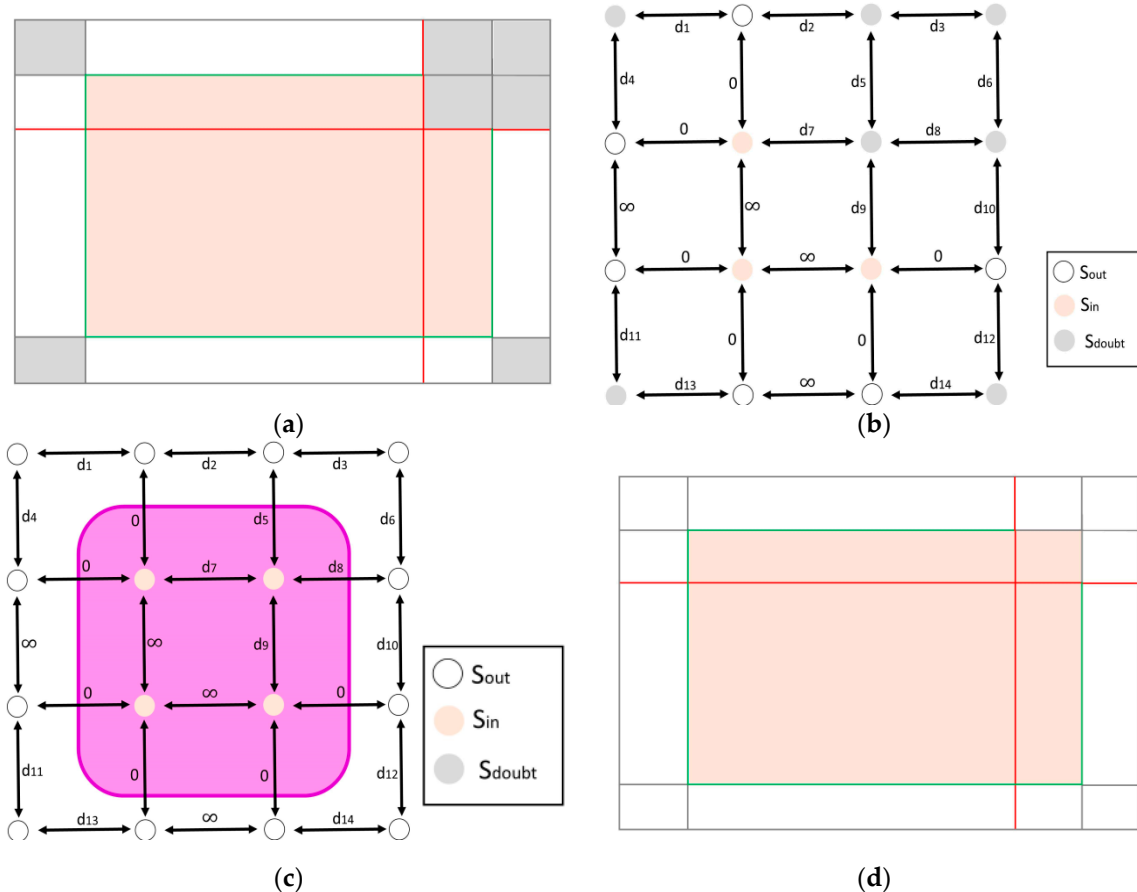
If the cell is classified as empty starting from ceiling occupancy:

$$D_i(l_i) = \begin{cases} 0 & l_i \in ext \\ \infty & l_i \in int \end{cases} \tag{8}$$

Otherwise:

$$D_i(l_i) = \begin{cases} \frac{i_{num}}{max_{num}} \cdot s_{\alpha} & l_i \in ext \\ \left(1 - \frac{i_{num}}{max_{num}}\right) \cdot s_{\alpha} & l_i \in int \end{cases} \quad (9)$$

where  $i_{num}$  is the intersection number of each cell,  $max_{num}$  is the maximum number of intersections in all cells and  $s_{\alpha}$  is the reciprocal of each cell's area.



**Figure 5.** Example of indoor component reconstruction using ‘s–t cut’ for Room 1: (a) initial labelling considering ceiling occupancy, (b) graph construction and cost assignment; (c) final binary labeling with minimum cost; and (d) final complex labeling.

### 3.4.2. Smoothing Term

A weight  $W_{i,j}$  is defined to determine the weights of adjacent cells, denoting the shared edges between the connected cells. By using the previous classification between occupied and empty cells the weights of edges between two occupied cells and between two empty are set to infinite and the weights of edges connecting an empty and an occupied cell are set to zero. In this way, cells forming the inner part of the room are prevented from being erroneously labelled as empty or vice-versa. The weights of the remaining edges between cells are fixed equal to the length of the edge between the cells. This means that the s – t cut problem is aimed at minimizing the length of guessed walls segments (Figure 5b).

$$W_{i,j} = S_{i,j}(l_i, l_j) = \begin{cases} 0 & \text{if } (l_i = occupied \text{ AND } l_j = occupied) \text{ OR } (l_i = empty \text{ AND } l_j = empty) \\ \infty & \text{if } (l_i = empty \text{ AND } l_j = occupied) \text{ OR } (l_i = occupied \text{ AND } l_j = empty) \\ LE_{ij} & \text{other} \end{cases} \quad (10)$$

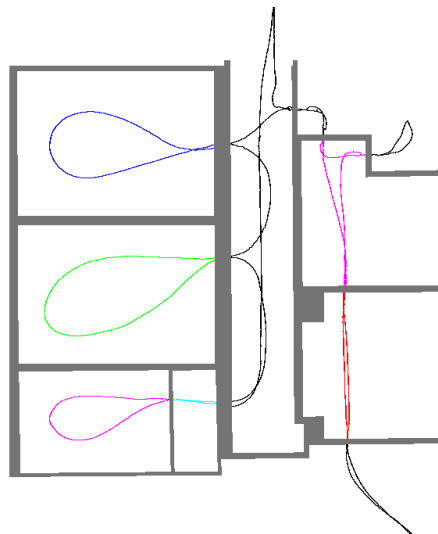
To perform the  $s - t$  cut, the Kolmogorov's max-flow algorithms is used [40]. Once having computed the  $S, T$  partitioning, the boundary of the occupied cells of the polygon partition gives the floor plan (Figure 5c,d).

### 3.5. Element Classification and Opening Detection

Once all of the indoor structure is determined with wall elements the presence of the openings is investigated. Detecting the boundaries of openings, such as windows or doors in a wall is a complex task. While in façade reconstruction applications, windows are generally detected as holes in the facade point cloud [42], this does not generally hold for the indoor environment. Indeed, also occlusions and clutter produce significant holes in the point cloud which have to be distinguished from real openings. To this end ray-tracing labelling is performed and an occupancy map is generated [23].

In this step each room and more specifically each surface is separately processed. Once having previously defined the wall surface, points representing inliers for the defined plane can be easily recognized. The detected plane is then discretized into cells of size  $\beta_2 \times \beta_2$  and then an occupancy map (denoted as OM) is generated on the basis of whether inlier points are detected at each pixel location or not. Without additional information, it is not possible to distinguish between a pixel that is truly empty and one that is merely occluded. This problem can be solved by using ray-tracing labelling to detect occlusion between the sensor and the analyzed surface. For this reason the scanning locations (position) should be known.

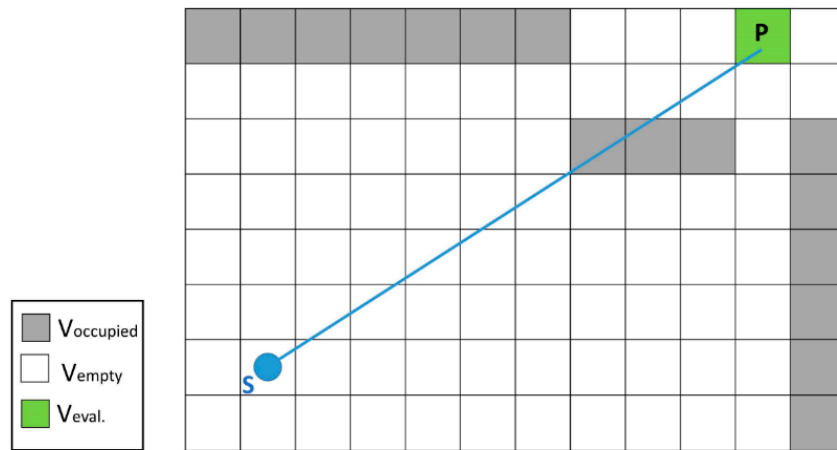
In particular, starting from the mobile scanner trajectory  $TR = \{trk \mid k \in \{1, \dots, M\}\}$  it is easily possible to identify the scanning points belonging to each room (Figure 6).



**Figure 6.** Subdivision of the acquisition path into sub-paths for each room (points belonging to each room are colored in a different way).

Let  $P = \{P_1, P_2, \dots, P_N\}$  be the set of scanned point for the room to be modelled. For each scan position  $S_K$  a labelling  $L_K$  is generated by tracing a ray from the scan location to each pixel  $P_i(x,y,z)$  labeled as 'empty' in the occupancy map (OM).

Having defined cell location and the scan location it is possible to verify if the defined ray is traversing a voxel (generated in the point cloud consolidation step) that is labelled as occupied. If so this means that  $P_i$  is occluded by some points in the scan and the cell is consequently labeled (Figure 7). On the other hand, if the ray is not traversing any occupied voxel the cell  $P_i$  is recognized as a real empty area.



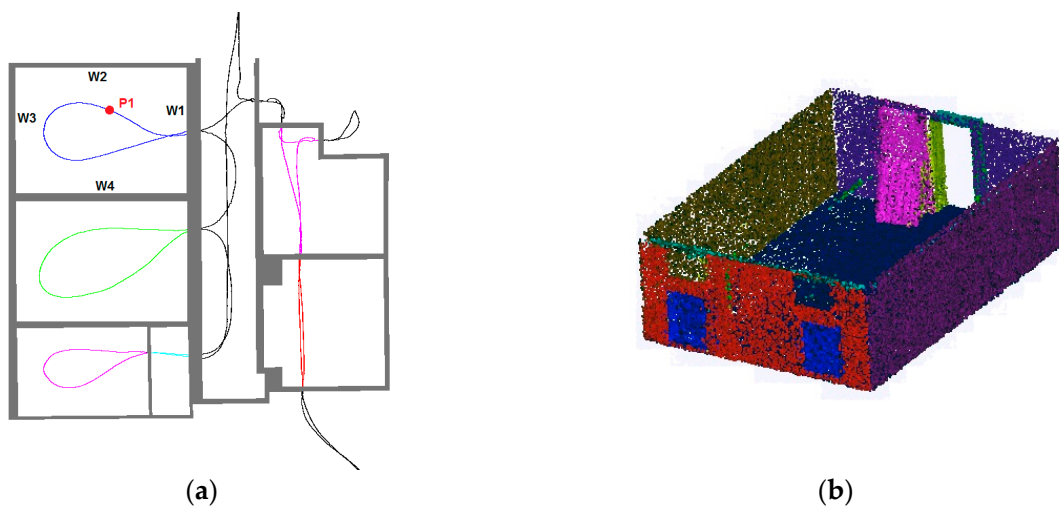
**Figure 7.** Ray-tracing labelling principle: point  $P$  is marked as occluded because the ray connecting it to the scanner position  $S$  traverses a voxel that is defined as occupied.

For the discretization of the voxel space a cylindrical buffer of three voxels is considered around the identified ray. After this ray-tracing labelling for all scan positions,  $K$  labels for each pixel are obtained. Figure 8 presents an example of OM generation for a specific position  $P_1$  in Room 1 (Figure 8a). As can be appreciated from Figure 8b, the door and another piece of furniture (pink segment) occludes a portion of the wall close to the entrance (Wall 1). In a similar way radiators (blue segment) occlude a portion of the wall under the windows (Wall 3). Starting from position  $P_1$  OMs are generated for all the four detected walls of the room (Figure 8c–f). In particular, taking into consideration Wall 1 (Figure 8c) and Wall 3 (Figure 8e) it is possible to notice the classification of occluded areas (in red) and of openings (in green).

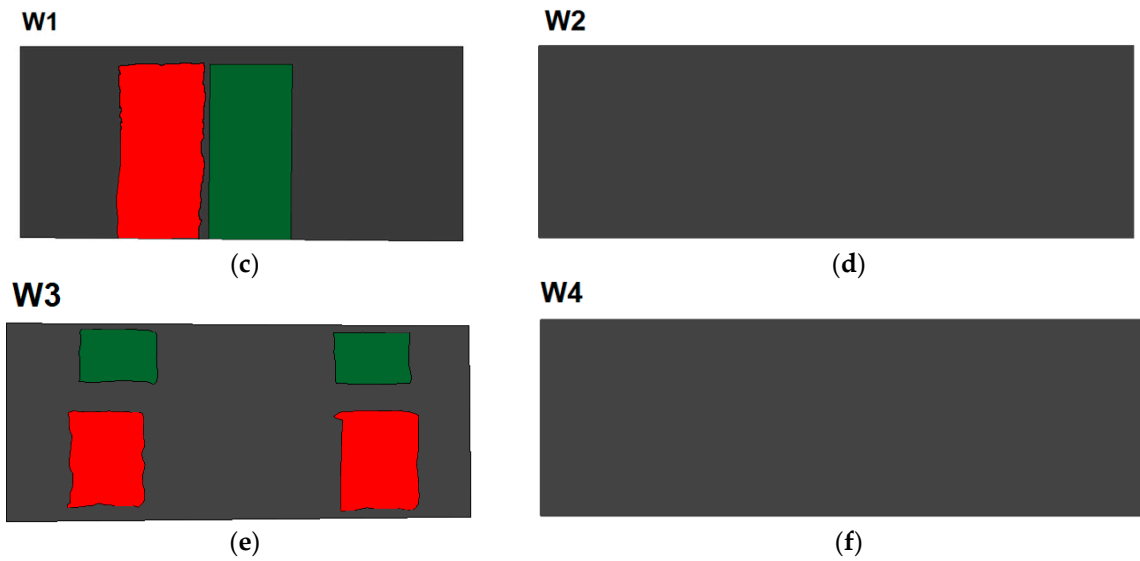
Finally, all the labels are combined in a final occupancy map (LF) adopting the following labeling rule:

$$\text{If } L_0 = \text{empty and } L_j = \text{occluded}, \forall j = 1, 2, \dots, K \Rightarrow L_F(i) = \text{occluded} \quad (11)$$

In other words, a cell is considered occluded if it is occluded from every scan-point.



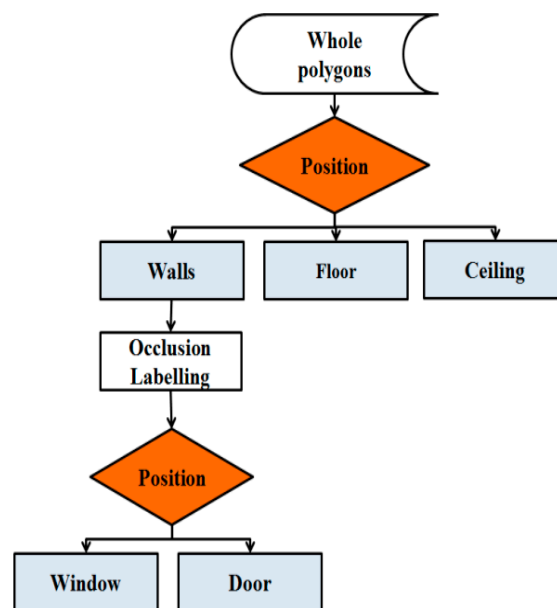
**Figure 8.** Cont.



**Figure 8.** Reconstruction of openings: (a) position  $P_1$  of the MLS sensor used to compute the occupancy map for Room 1; (b) segmented point cloud for Room 1 (ceiling is removed to look into the room); and (c–f) occupancy maps generated for the four walls in Room 1 (in gray—occupied cells, in red—occluded cells, and in green—openings).

### 3.6. Element Classification and Model Generation

A procedure similar to the one described in [33] is used to classify openings into windows and doors as well as to recognize their shape. In particular, a hierarchical classification tree is used (Figure 9). Openings are classified as doors when they intersect with the ground floor. Once the raw shape of the openings is determined priors on indoor architecture are added to generate the room model. In particular, the prevalence in building rooms of straight lines and orthogonal intersection is exploited to add additional constraints to enforce the modeling.



**Figure 9.** Hierarchical classification tree, orange diamonds are the conditions while blue rectangles represent room elements.



#### 4. Applications and Accuracy Evaluation

This section presents the results of the adopted method on the data set used in this paper and discusses the parameter selection.

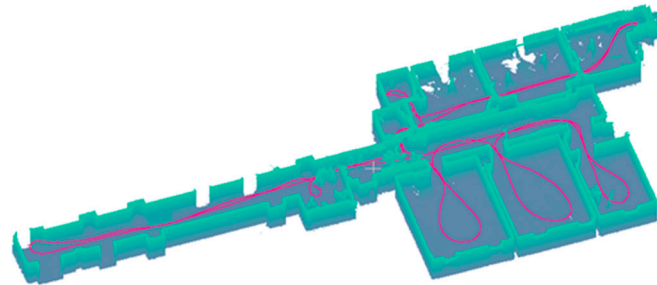
##### 4.1. Data Sets

The methodology is tested in a real case study: an academic building indoors surveyed with the Indoor Modelling System Viametris IMS3D. The dataset is provided by the ISPRS Scientific Initiative on Indoor Modelling [12] and the technical characteristics of the laser device are summarized in Table 1.

**Table 1.** Technical characteristics of the Viametris IMS3D laser scanning device according to the manufacturer's datasheet.

Technical Characteristics	
Maximum measurement range	80 m
Data acquisition rate	$600 \cdot 10^3$ points/s
Resolution	0.125° horizontal, 0.125° vertical
Angular Field of View (FoV)	360° × 360°
Relative accuracy	30 mm (0.1 to 10 m)
Absolute position accuracy	2 cm

The dataset corresponds to an acquisition carried out in a building of the Technische Universität Braunschweig (Germany) and it includes both the point cloud and the trajectory followed by the system during data acquisition (Figure 10). The complete indoor scene comprises 10 rooms belonging to the same floor although in this work just a set of seven rooms were processed.



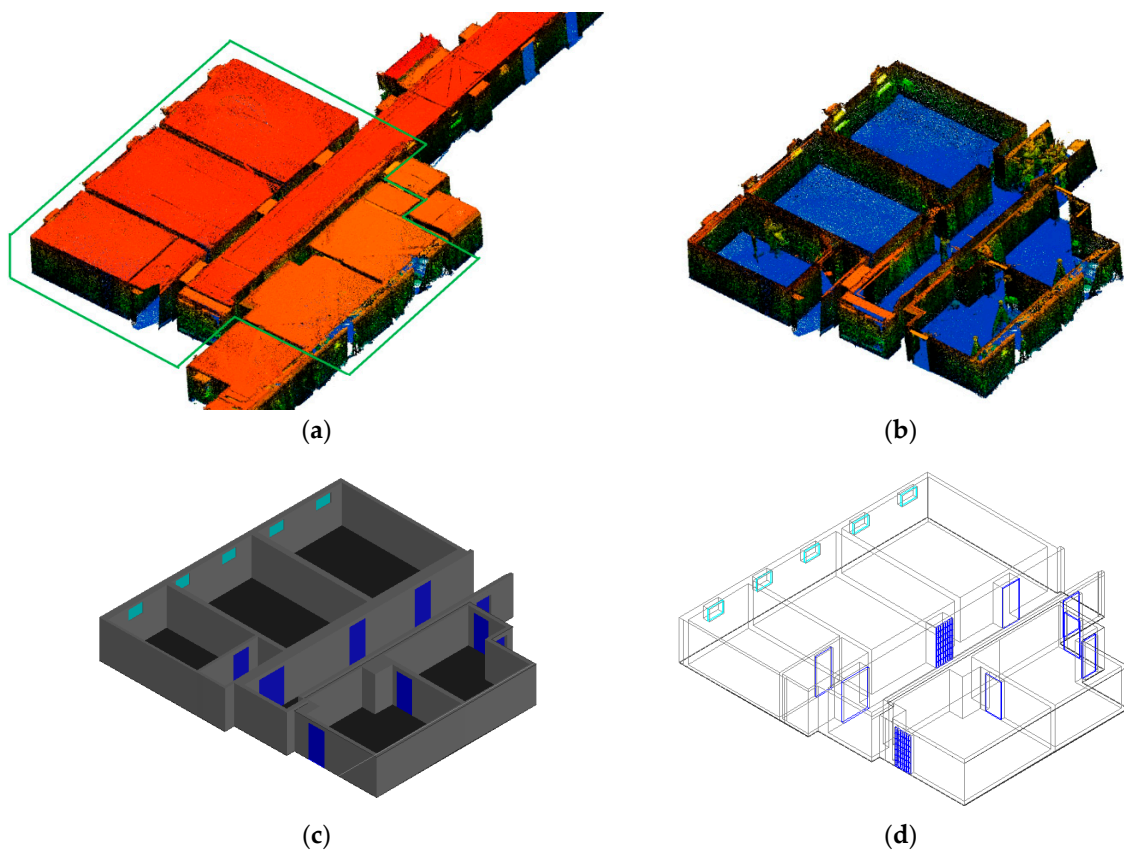
**Figure 10.** The data set provided by the ISPRS Scientific Initiative on Indoor modelling used as case study.

##### 4.2. Results

Figure 11 shows the reconstruction results for the data sets used. It is worth noting that all rooms were successfully reconstructed and all doors and windows detected except one window that was almost completely occluded by curtains, which is in conflict with the assumption of this method.

The list of parameters controlling the data processing are reported in Table 2. They may be categorized into four groups. The first one is mainly addresses point cloud smoothing and regularization. Indeed, even if the RANSAC extraction method is quite robust against noise and outliers the enhancement of the point cloud simplifies and speeds up the plan identification phase. In particular, the principal parameters in this step (and also of the element classification and occlusion labelling step) is the choice of the voxel grid spacing ( $V_R$ ). The optimal  $V_R$ , as previously discussed, is a function of the mean sampling resolution of the point cloud and in particular, it should be slightly larger with respect to the average distance between points in the point cloud. Indeed, shorter spacing will produce computational overload and will determine the generation of a large set of disconnected components. On the other hand, larger spacing would result in a loss of the accuracy smoothing effect.

In particular, for the data set adopted in the experiments,  $V_R$  was set 1.5 times the mean point spacing. A similar consideration can be formulated for parameters  $\beta_1$  and  $\beta_2$ .



**Figure 11.** Experimental results: (a) the input point cloud and the analyzed area (green boundaries); (b) a detail of the analyzed area without the ceiling; (c) the reconstructed model without ceiling; and (d) a wireframe view of the model.

**Table 2.** Parameters adopted during data processing.

Parameter	Description
	Point cloud consolidation
$V_R$	Voxel grid spacing
$K$	Neighbor size for estimation of normal
$n_c$	Number of points in each voxel
	Point cloud segmentation
$\epsilon$	Bandwidth for the RANSAC based plan extraction (i.e., maximum distance between inliers points and the estimated plan in the RANSAC algorithm)
$\alpha$	Maximum distance between local point normal and plan normal in the estimated plan during RANSAC processing
$\beta_1$	Cell size to generate binary image
$\tau_R$	Dominant line threshold
$n_R$	Minimum number of inlier points in RANSAC primitive
	Indoor component reconstruction
$\lambda$	Weight balancing the data term and the smoothness term
	Element classification and occlusion labelling
$\beta_2$	Cell size to generate the occupancy map
$V_R$	Voxel grid spacing

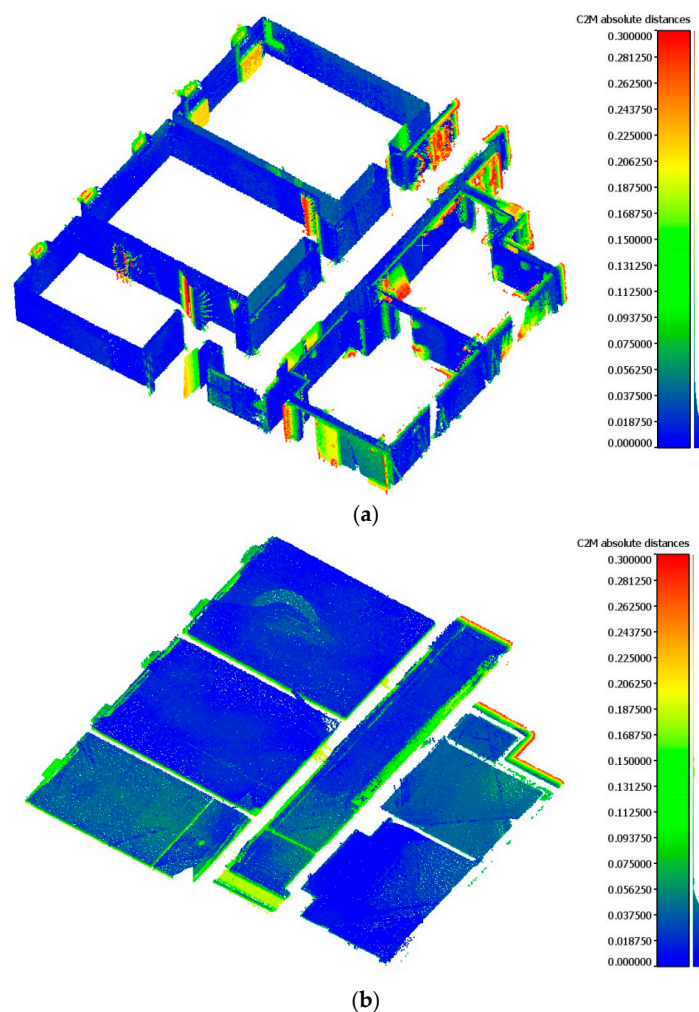
The second category of parameters is used to set point cloud processing and planar primitive extraction. Since the RANSAC plane detection is affected by noise and inaccuracy in normal estimation because of outliers and noise, parameters  $\epsilon$  and  $\alpha$  are designed to compensate for these possible effects. The parameter  $\tau_R$  governs the fusion of similar plans extracted by RANSAC and thus it is indeed designed to mitigate for over-segmentation problems. Indeed, an over-segmented point cloud is likely to be more prone to classification errors. In the presented example,  $\tau_R = 2\epsilon$  was selected for the plans distance and  $\tau_R = \alpha$  for normal distances.

The parameter  $\lambda$  weights the data term and the smoothness term in Equation (6). To test the contribution of these two terms in the final model, different values of  $\lambda$  were tested. For lower values of  $\lambda$ , the smoothness term increased in favor of a smoother solution and consequently some details were filtered out (Figure 12). In the presented case, a value of  $\lambda$  ranging between 0.5 and 0.8 provided the best reconstructed model.

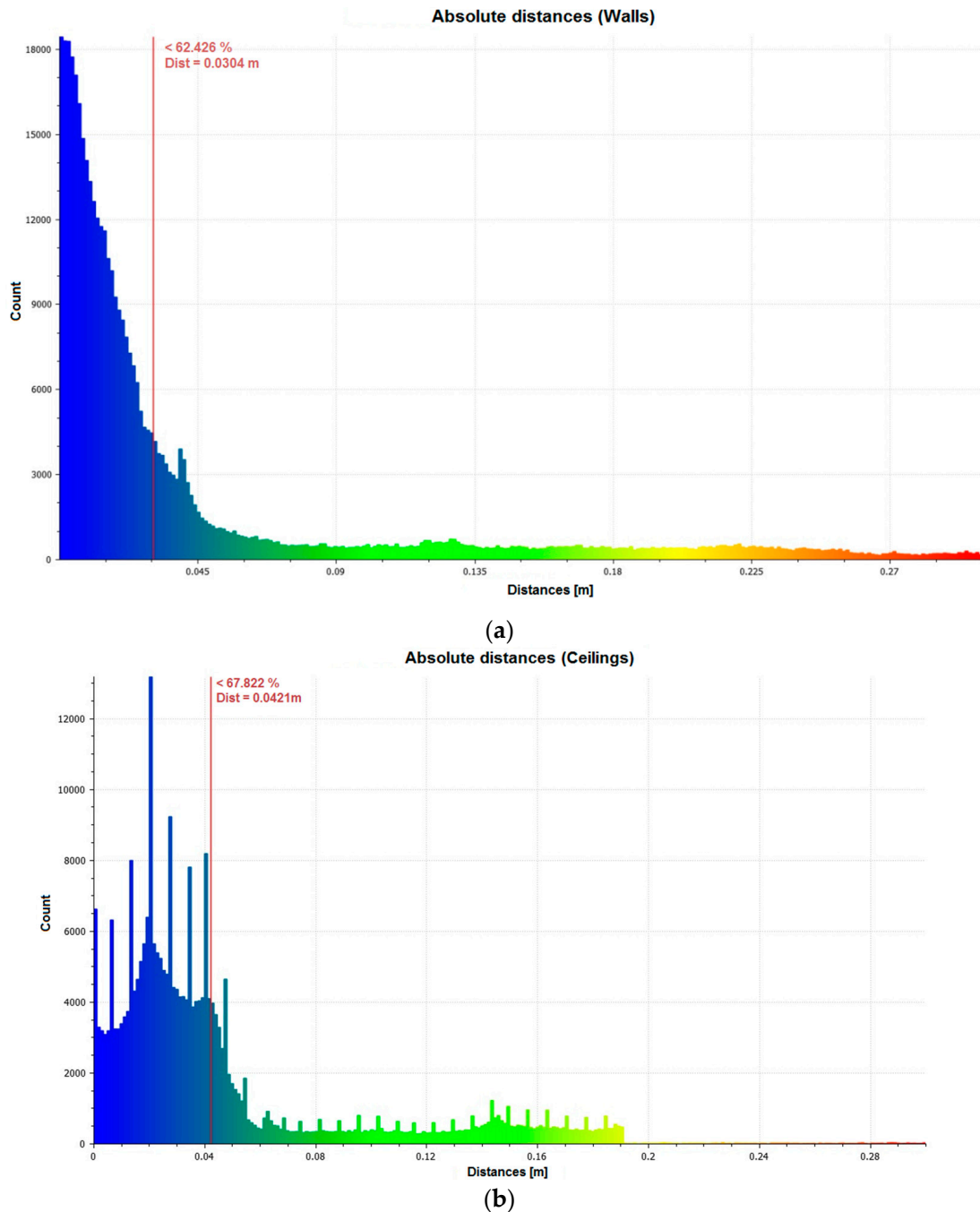


**Figure 12.** ‘Cell complex’ reconstruction for the adopted data set: (a) detected primitives; (b) induced ‘cell complex’ with ‘pending walls’ (red); (c) initial labeling of the ‘cell complex’; (d) final labelling ( $\lambda = 0.8$ ); and a detail of the complex with different values of  $\lambda$  (e).

To evaluate the quality of the generated model the standard deviation of the unsigned distances between the original point cloud and the reconstructed objects (walls, floor, and ceiling) was used. Figure 13a,b presents point clouds colored with the absolute distance associated to the walls and ceiling, respectively. For the data set considered the RMS of the unsigned distance is slightly lower than 3.0 cm. This means that on average, the walls are reconstructed with a precision of about 3.0 cm (which can be considered comparable to the accuracy of the original point cloud). Figure 14a shows the histogram of residuals between the reconstructed model and point cloud for the walls, showing that more than 65% of the points has a distance from the reconstructed model smaller than 3.0 cm. The visualization of absolute distances (Figure 13a) thanks to a color-bar allows those walls or portions of walls having a larger discrepancy to the point cloud to be displayed rapidly. For example, a small portion of walls feature a standard deviation of 20 cm which is likely to mean that the reconstruction has been wrong or that the object is not really a portion of wall (e.g., it could be a piece of furniture). Indeed, assuming that there is no significant noise in the data, the factors that might influence the standard deviation of the models are the number of planes composing it and the amount of points which emerge from the wall (objects leaning against walls). Concerning the ceiling, a similar consideration can be adopted. However, the distribution of absolute distances follows an unusual path (Figure 14b). This is probably due to the fact that in the current implementation the case of a suspended ceiling (i.e., a ceiling with different heights in the same room) has not been considered. However, in the data set this situation occurred causing larger discrepancies in those areas, as in the yellow areas in Figure 13b.



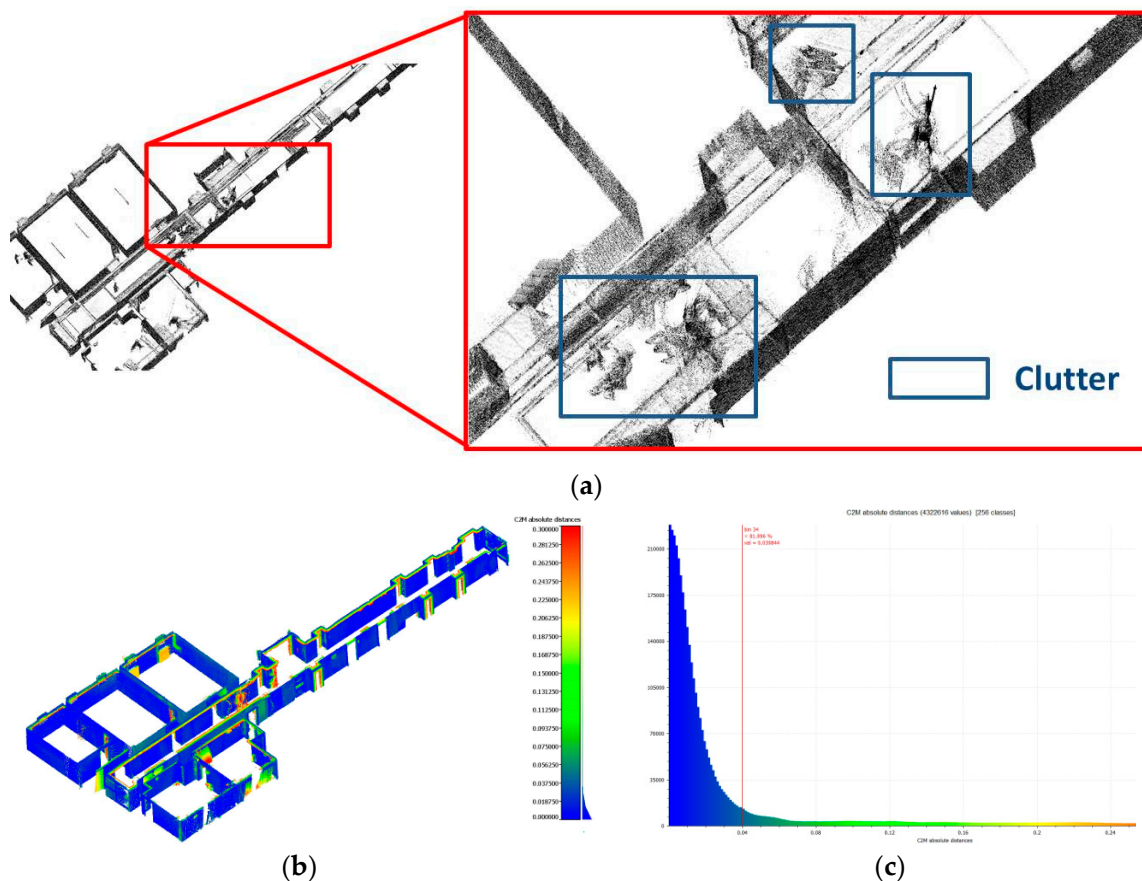
**Figure 13.** Point clouds colored with points-to-model unsigned distances for walls (a) and ceiling (b).



**Figure 14.** Histogram of residuals between the reconstructed model and point cloud for walls (a) and ceiling (b).

A second test was performed on an extended area, including the central corridor, of the same dataset. The aim of this second test was to verify the robustness of the presented method in the case of a large amount of clutter. Indeed, due to the presence of moving people during the data acquisition phase the scene was characterized by significant clutter (Figure 15a). Also in this case results in terms of data completeness and data quality confirm the ones previously obtained. In particular, more than 81% of points has a discrepancy from the reconstructed model which is smaller than 4.0 cm.





**Figure 15.** Corridor dataset: in the middle of the corridor due to the presence of people it presents a highly cluttered area (a), point clouds colored with points-to-model unsigned distances for walls (b), and histogram of residuals between the reconstructed model and point cloud for walls (c).

### 5. Conclusions

This paper presented a method for indoor building modeling starting from MLS; the core of the methodology is the transformation of the indoor reconstruction into the labeling problem of structural cells in a 2D floor plan. The input data of the presented method are point cloud and scanner trajectory. The method was designed to extract watertight 2D floor plans and reconstruct in a plausible way occlusion both in 2D floor plans and in 3D walls. The obtained model can also include semantic information enhancing in this way the capacity of the adopted methodology to classify recognized component such as walls, doors, ceilings, and windows. The method proved to be robust against noise and occlusions. In the experiments carried out, the final models presented accuracies comparable with the quality of the input point cloud.

A major limitation to the full optimization of the method is due to the Manhattan-World domain assumption. Because of such an assumption, walls which build up the floor plan are assumed to be rectangular and intersect orthogonally. However, more complex room geometries such as the curved surfaces of triangular walls (as in the case of lofts) may occur in scanning indoor environments. In addition, also “pending walls” are assumed to rely on the Manhattan-World assumption. The methodology also relies on the assumption of horizontal and planar ceilings and floors. In order to reduce model assumptions, in our future work we are planning to reconstruct indoor scenes also taking into consideration cylindrical walls and spherical or freeform ceilings by using a hybrid indoor model representation combining mesh and geometrical primitives. Since the method is performed off-line, it cannot be used for real time localization

**Author Contributions:** Conceptualization, M.P. and L.D.-V.; Methodology, M.P.; Data curation, L.D.-V., M.S. coordinated and reviewed the manuscript.

**Acknowledgments:** The second author would like to give thanks to the Xunta de Galicia for the financial support given through human resources grant no ED 481B 426 2016/079-0.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Diaz-Vilarino, L.; Boguslawski, P.; Khoshelham, K.; Lorenzo, H.; Mahdjoubi, L. Indoor navigation from point clouds: 3D modelling and obstacle detection. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *41*, 275–281. [[CrossRef](#)]
2. Liu, L.; Zlatanova, S. Generating navigation models from existing building data. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2013**, *40*, 19–25. [[CrossRef](#)]
3. Díaz-Vilariño, L.; Lagüela, S.; Armesto, J.; Arias, P. Indoor daylight simulation performed on automatically generated as-built 3D models. *Energy Build.* **2014**, *68*, 54–62. [[CrossRef](#)]
4. Zlatanova, S.; Sithole, G.; Nakagawa, M.; Zhu, Q. Problems in indoor mapping and modelling. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2013**, *40*, 63–68. [[CrossRef](#)]
5. Puente, I.; Arias, P. Review of mobile mapping and surveying technologies. *Measurement* **2013**, *46*, 2127–2145. [[CrossRef](#)]
6. Remondino, F.; Nocerino, E.; Toschi, I.; Menna, F. A critical review of automated photogrammetric processing of large datasets. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, *42*. [[CrossRef](#)]
7. Díaz-Vilariño, L.; Verbree, E.; Zlatanova, S.; Diakité, A. Indoor modelling from SLAM-based laser scanner: Door detection to envelope reconstruction. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, *42*, 345–352. [[CrossRef](#)]
8. Gashongore, P.D.; Kawasue, K.; Yoshida, K.; Aoki, R. Indoor space 3D visual reconstruction using mobile cart with laser scanner and cameras. In Proceedings of the Eighth International Conference on Graphic and Image Processing, Tokyo, Japan, 29–31 October 2016; Volume 10225, p. 102251E.
9. Nocerino, E.; Menna, F.; Remondino, F.; Toschi, I. Investigation of indoor and outdoor performance of two portable mobile mapping systems Investigation of indoor and outdoor performance of two portable mobile mapping systems. In Proceedings of the SPIE Optical Metrology, Munich, Germany, 25–29 June 2017; Volume 10332, p. 10332201.
10. Tucci, G.; Visintini, D.; Bonora, V.; Parisi, E.I. Examination of Indoor Mobile Mapping Systems in a Diversified Internal/External Test Field. *Appl. Sci.* **2018**, *8*, 401. [[CrossRef](#)]
11. Thrun, S. *Simultaneous Localization and Mapping*; Springer: Berlin, Germany, 2007; pp. 13–14.
12. Khoshelham, K.; Vilariño, L.D.; Peter, M.; Kang, Z.; Acharya, D. The ISPRS benchmark on indoor modelling. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, *42*, 367–372. [[CrossRef](#)]
13. Oesau, S.; Lafarge, F.; Alliez, P. Indoor scene reconstruction using feature sensitive primitive extraction and graph-cut. *ISPRS J. Photogramm. Remote Sens.* **2014**, *90*, 68–82. [[CrossRef](#)]
14. Ochmann, S.; Vock, R.; Wessel, R.; Klein, R. Automatic reconstruction of parametric building models from indoor point clouds. *Comput. Graph.* **2016**, *54*, 94–103. [[CrossRef](#)]
15. Sanchez, V.; Zakhor, A. Planar 3D modeling of building interiors from point cloud data. In Proceedings of the 2012 19th IEEE International Conference on Image Processing, Orlando, FL, USA, 30 September–3 October 2012; pp. 1777–1780.
16. Hähnel, D.; Burgard, W.; Thurn, S. Learning compact 3D models of indoor and outdoor environments with a mobile robot. *Rob. Auton. Syst.* **2003**, *44*, 15–27. [[CrossRef](#)]
17. Budroni, A.; Boehm, J. Automated 3D Reconstruction of Interiors from Point Clouds. *Int. J. Archit. Comput.* **2010**, *8*, 55–74. [[CrossRef](#)]
18. Nüchter, A.; Hertzberg, J. Towards semantic maps for mobile robots. *Rob. Auton. Syst.* **2008**, *56*, 915–926. [[CrossRef](#)]
19. Furukawa, Y.; Curless, B.; Seitz, S.M.; Szeliski, R. Reconstructing building interiors from images. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision, Kyoto, Japan, 29 September–2 October 2009; pp. 80–87.

20. Khoshelham, K.; Díaz-Vilariño, L. 3D modelling of interior spaces: Learning the language of indoor architecture. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2014**, *40*, 321–326. [[CrossRef](#)]
21. Tran, H.; Khoshelham, K.; Kealy, A.; Díaz-Vilariño, L. Extracting topological relations between indoor spaces from point clouds. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, *4*, 401–406. [[CrossRef](#)]
22. Li, C.; Li, G.; Deng, Y.; Wang, T.; Kang, Z. Research on knowledge-based optimization method of indoor location based on low energy Bluetooth. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, *42*. [[CrossRef](#)]
23. Adan, A.; Huber, D. 3D reconstruction of interior wall surfaces under occlusion and clutter. In Proceedings of the 2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission, Hangzhou, China, 16–19 May 2011; pp. 275–281.
24. Previtali, M.; Barazzetti, L.; Brumana, R.; Scaioni, M. Towards automatic indoor reconstruction of cluttered building rooms from point clouds. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2014**, *2*, 281–288. [[CrossRef](#)]
25. Díaz-Vilariño, L.; Khoshelham, K.; Martínez-Sánchez, J.; Arias, P. 3D modeling of building indoor spaces and closed doors from imagery and point clouds. *Sensors* **2015**, *15*, 3491–3512. [[CrossRef](#)] [[PubMed](#)]
26. Díaz-Vilariño, L.; Martínez-Sánchez, J.; Lagüela, S.; Armesto, J.; Khoshelham, K. Door recognition in cluttered building interiors using imagery and LiDAR data. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2014**, *40*, 203–209. [[CrossRef](#)]
27. Quintana, B.; Prieto, S.A.; Adán, A.; Bosché, F. Door detection in 3D colored laser scans for autonomous indoor navigation. In Proceedings of the 2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Alcalá de Henares, Spain, 4–7 October 2016; pp. 4–7.
28. Nikoohemat, S.; Peter, M.; Oude Elberink, S.; Vosselman, G. Exploiting Indoor Mobile Laser Scanner Trajectories for Semantic Interpretation of Point Clouds. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, *4*, 355–362. [[CrossRef](#)]
29. Förstner, W. Optimal vanishing point detection and rotation estimation of single images from a legoland scene. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2010**, *38*, 157–162.
30. Alsadik, B.; Gerke, M.; Vosselman, G. Visibility analysis of point cloud in close range photogrammetry. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2014**, *2*, 9–16. [[CrossRef](#)]
31. Gröger, G.; Plümer, L. CityGML-Interoperable semantic 3D city models. *ISPRS J. Photogramm. Remote Sens.* **2012**, *71*, 12–33. [[CrossRef](#)]
32. Previtali, M.; Barazzetti, L.; Brumana, R.; Cuca, B.; Oreni, D.; Roncoroni, F.; Scaioni, M. Automatic façade modelling using point cloud data for energy-efficient retrofitting. *Appl. Geomat.* **2014**, *6*, 95–113. [[CrossRef](#)]
33. Previtali, M.; Scaioni, M.; Barazzetti, L.; Brumana, R.; Roncoroni, F. Automated Detection of Repeated Structures in Building Facades. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2013**, *2*, 241–246. [[CrossRef](#)]
34. Fischler, M.A.; Bolles, R.C. Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography. *Commun. ACM* **1981**, *24*, 381–395. [[CrossRef](#)]
35. Comaniciu, D.; Meer, P. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 603–619. [[CrossRef](#)]
36. Chauve, A.L.; Labatut, P.; Pons, J.P. Robust piecewise-planar 3D reconstruction and completion from large-scale unstructured point data. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 1261–1268.
37. Edelsbrunner, H.; O'Rourke, J.; Seidel, R. Constructing arrangements of lines and hyperplanes with applications. *SIAM J. Comput.* **1986**, *15*, 341–363. [[CrossRef](#)]
38. Lempitsky, V.; Blake, A. LogCut—Efficient Graph Cut Optimization for Markov Random Fields. In Proceedings of the 2007 11th International Conference on Computer Vision, Rio de Janeiro, Brazil, 14–21 October 2007; pp. 1–8.
39. Kada, M.; Luo, F. Generalisation of building ground plans using half-spaces. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2006**, *36*, 2–5.
40. Boykov, Y.; Kolmogorov, V. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 1124–1137. [[CrossRef](#)] [[PubMed](#)]

41. Wang, R.; Xie, L.; Chen, D. Modeling indoor spaces using decomposition and reconstruction of structural elements. *Photogramm. Eng. Remote Sens.* **2017**, *83*, 827–841. [[CrossRef](#)]
42. Boulaassal, H.; Landes, T.; Grussenmeyer, P. Automatic extraction of planar clusters and their contours on building façades recorded by terrestrial laser scanner. *Int. J. Archit. Comput.* **2009**, *7*, 1–20. [[CrossRef](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).