

Article

The Polygonal 3D Layout Reconstruction of an Indoor Environment via Voxel-Based Room Segmentation and Space Partition

Fan Yang ^{1,2} , You Li ^{3,4}, Mingliang Che ¹, Shihua Wang ^{1,5}, Yingli Wang ¹, Jiye Zhang ¹, Xinliang Cao ¹ and Chi Zhang ^{1,*}

- ¹ Nantong Key Laboratory of Spatial Information Technology R&D and Application, School of Geographic Science, Nantong University, Nantong 226019, China
² Key Laboratory of Urban Land Resources Monitoring and Simulation, Ministry of Natural Resources, Shenzhen 518034, China
³ Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen 518107, China
⁴ Smart Cities, School of Architecture and Urban Planning, Shenzhen University, Shenzhen 518060, China
⁵ Hai'an Huajun Measurement Co., Ltd., Hai'an 226602, China
* Correspondence: zhangchi1982@ntu.edu.cn; Tel.: +86-13645161068

Abstract: An increasing number of applications require the accurate 3D layout reconstruction of indoor environments. Various devices including laser scanners and color and depth (RGB-D) cameras can be used for this purpose and provide abundant and highly precise data sources. However, due to indoor environment complexity, existing noise and occlusions caused by clutter in acquired data, current studies often require the idealization of the architecture space or add an implication hypothesis to input data as priors, which limits the use of these methods for general purposes. In this study, we propose a general 3D layout reconstruction method for indoor environments. The method combines voxel-based room segmentation and space partition to build optimum polygonal models. It releases idealization of the architectural space into a non-Manhattan world and can accommodate various types of input data sources, including both point clouds and meshes. A total of four point cloud datasets, four mesh datasets and two cross-floor datasets were used in experiments. The results exhibit more than 80% completeness and correctness as well as high accuracy.

Keywords: 3D layout; indoor space; voxel; space partition; point cloud; mesh



Citation: Yang, F.; Li, Y.; Che, M.; Wang, S.; Wang, Y.; Zhang, J.; Cao, X.; Zhang, C. The Polygonal 3D Layout Reconstruction of an Indoor Environment via Voxel-Based Room Segmentation and Space Partition. *ISPRS Int. J. Geo-Inf.* **2022**, *11*, 530. <https://doi.org/10.3390/ijgi11100530>

Academic Editors: Wolfgang Kainz and Sisi Zlatanova

Received: 16 August 2022

Accepted: 16 October 2022

Published: 19 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 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 (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The three-dimensional (3D) layout reconstruction of indoor environments is of great significance for many fields, such as games, films, intelligent buildings, building information model (BIM) and robotics [1–4]. In the game and film industry, an indoor 3D layout of the environment is helpful for the integration of the real-world and virtual information and for realizing augmented reality (AR) and mixed reality (MR) [2]. In the field of architecture, engineering and construction (AEC), indoor structure and layout are important components of BIM [5]. Digital representations of urban architecture are becoming increasingly important due to the increasing growth of urban buildings [6]. They are inseparable from the meta-universe and digital twin cities.

Indoor 3D layout reconstruction is used to build 3D models integrating points, lines and surfaces. Spatial layouts provide a higher level of knowledge of architectural space [7]. Image-based photogrammetry technology and laser scanning-based point cloud data acquisition technology provide effective data sources for indoor spatial information acquisition and layout information extraction [8]. To meet the demands of various applications, both laser point clouds and images are crucial data sources, and their respective advantages are combined to maximize the availability of information. However, the laser point clouds or meshes obtained are usually unstructured and lack semantic information. The generalized

point cloud data obtained contain noise, occlusion and incomplete data because of the complicated indoor structure and the enormous number of items accumulating in the space. Therefore, the construction of an urban information model is still time-consuming, laborious and expensive.

Despite the heavy demand for digital buildings, the work to build accurate, up-to-date and semantic-rich building models is still a difficult task. Points are obtained along the laser scanner's line of sight when data are being acquired. The generated mesh model or point cloud data suffer from noise and data loss due to occlusion brought on by indoor items and the various reflective qualities of different types of objects. Existing methods have difficulty effectively building indoor 3D models with semantic and topological consistency. Many scholars simplify the problem from the perspective of architecture space structure, limiting the building space to a 2.5-dimensional (2.5D) idealization or adopting a more rigorous Manhattan world assumption. Other scholars have studied rapid construction methods of building models, where prior knowledge is added to the data acquisition process, such as a laser scan for each room, and the trajectory forms an aggregation in the room [9]. Artificially adding prior knowledge to data increases the manual workload, and it cannot meet the needs of the next generation of robot intelligence. In addition, none of the current methods can solve the problem of cross-floor space reconstruction. Therefore, a general indoor layout reconstruction method is still missing [1].

In this study, a general 3D layout reconstruction method combining voxel-based room segmentation and space partitioning is proposed. The main contribution is creating polygonal structural descriptions of buildings from laser scanners and camera image data for other than simple Manhattan geometry. The method associates room semantic information to divide subspaces to acquire optimum polygonal models of indoor environments. It provides 3D layout reconstructions of multiple rooms, multiple stories and a non-Manhattan world, especially for the reconstruction of the cross-floor space. The data input can be point cloud data or mesh data, with or without pose/viewpoint/trajectory information.

The paper is structured as follows: The associated research for this study is introduced in Section 2, the proposed reconstruction method is introduced in Section 3, the experiments are described in Section 4, the discussion is presented in Section 5, and the conclusions are presented in Section 6.

2. Related Works

A growing interest in indoor 3D reconstruction has been sparked by the advancement of laser scanning and photogrammetry technology. In the computer vision field, many studies focus on building a surface visualization 3D model, and all objects in the reconstructed scene are represented by a triangular mesh. Commonly used scene surface reconstruction methods include Delaunay triangulation [10] and Poisson reconstruction [11]. However, the surface model lacks structural and semantic information and cannot meet the needs of deep-seated analysis and calculation.

The interior layout provides high-level information on the interior space structure, which is of great significance for virtual reality, interior design, navigation path planning and energy calculation. Many scholars have studied the estimation and reconstruction of indoor layouts [5]. According to the different data inputs, reconstruction can be divided into laser scanning-based [7] and image-based reconstruction [12], and the images include monocular images [13], panoramic images [14] and RGB-D images [15]. According to different scenes, the indoor layout can be classified as single rooms [14], multiple rooms [16] and large-scale indoor spaces [17]. The dimensionality can be two-dimensional (2D) [7,18] or 3D [19]. Significant trends are that the space range of room layout estimation is increasing, the indoor structure is becoming increasingly complex and the types of input data used are becoming increasingly diverse.

Rule-based methods and data-driven methods are the two groups into which the 3D layout reconstruction techniques may be divided. Rule-based methods manually use definitions of grammar rules, application orders and parameters, which are also known

as grammatical information in the 3D reconstruction process. A two-rule grammar was suggested by Khoshelham and Diaz-Vilarino [20] to reconstruct 3D interior spaces of Manhattan world buildings from point clouds. One of the rules is the absolute orientation of the main walls and the floors/ceilings. The other is that neighboring cuboids are connected when they are not separated by an interior wall. Becker et al. [21] used a grammar-based method that embeds the reconstruction process into an automatic learning and verification loop to reconstruct the 3D layout of building interiors from raw point clouds. Ikehata et al. [22] applied grammar rules to recover a structure graph together with geometries and segmented rooms with a heuristics algorithm. Murali et al. [23] implicitly used rules in which axis-aligned cuboids are connected. The method checks the intersections of detected planes and creates a connected wall graph. The cuboids are constructed after cycles are found in the wall graph. Subsequently, the cuboids are clustered together to form rooms. However, because extracting usable shape grammars for irregular structures (e.g., with slanted walls or with walls intersecting at random angles) is extremely difficult, these approaches are adopted mainly for regular Manhattan-world buildings.

When compared with rule-based methods, data-driven methods for indoor layout reconstruction are more susceptible to imperfect data [24]. These methods are commonly used for the 3D layout reconstruction of building interiors. The buildings have varying and complex structures, and many objects, such as furniture, exist in the room. During the process of image-based visual camera and laser scanning, due to the differences in line-of-sight obstruction caused by indoor objects and the reflection characteristics of different types of objects, the noise, obstruction and incomplete broad point cloud data obtained bring many difficulties to the estimation of indoor 3D layouts. To simplify the problem, many studies have imposed assumptions about architecture space or manually added experience-based information to data acquisition processes to reconstruct the building interior layout. The abstractions vary from the 2.5D space assumption and rigorous Manhattan-world assumption to single-story procedures. As the z-axis orientation is basic and adopted in most studies, we do not separately treat it as an assumption. Then, the 3D layout of the indoor environment is solved by dividing the indoor space according to the cellular characteristics of the space (such as room).

The rigorous Manhattan-world [21] hypothesis assumes that a plane is parallel to one of the three principal planes of the orthogonal coordinate system. There are only two possible relationships between the wall and the wall at this point: parallel and orthogonal. The wall is perpendicular to the floor. Armeni et al. [17] assumed that rooms were aligned to the Manhattan-world frame and parsed large-scale 3D point clouds of buildings into rooms. Building components such as walls, doors and objects were further categorized. To overcome the limitations of the Manhattan-world hypothesis, some scholars have adopted the weak Manhattan-world hypothesis. The weak Manhattan world assumes that the wall is vertical to the floor and that the vertical plane can be arbitrarily oriented around the vertical direction, which can achieve indoor 3D modeling in the case of nonorthogonal walls [25]. The 3D point clouds are projected to the XOY plane to realize 2D room segmentation. The 2.5D assumption is similar to the weak Manhattan-world hypothesis that was employed to model environments with vertical walls and horizontal floors and ceilings. However, the 2.5D assumption cannot be applied to the indoor environment of cross-floor space and nested rooms (rooms inside room), which limits room reconstruction in the non-Manhattan world.

Mura et al. [26] automatically selected the number of room clusters by grouping the viewpoint cells according to their visibility overlaps. This method overcomes the limitations of the 2.5D assumption and allows for the modeling of slanted wall and sloped ceiling structures. However, this ability is obtained by utilizing an artificial prior throughout the data collecting process, i.e., assuming that every room has at least one scan position. Ambrus et al. [27] computed a collection of simulated sensor poses/viewpoints that resulted in the preliminary labeling of the input point cloud.

Some researchers have applied trajectory information to the semantic classification of indoor space [9,28]. In contrast to terrestrial laser scanners (TLS), when a mobile laser scanner (MLS) is used for data acquisition, the perception of the space is continuous, and no separate scanning is performed for each room [29]. These methods assume that two places are connected by doors and subdivide indoor space into floors, stairs, porches and rooms. However, this method relies on a closed-loop strategy in the data acquisition process to form a trajectory cluster and thus a trajectory set in the room. When a room has more than one door and the distance between the trace points passing through the two doors is large, this method may fail.

Cui et al. [28] proposed a visibility analysis-based multiroom segmentation method. The visibility analysis is achieved by using ray tracing along the sampled trajectory points and the cells' center points of each detected patch. Similarly, Ochmann et al. [19] ran visibility tests between point patches on surfaces using ray casting and built a visibility graph. The regions of the point cloud with high mutual visibility formed clusters corresponding to the rooms of the building. However, the method is not so reliable, especially for cross-floor reconstruction.

Reconstructing indoor 3D models from 3D point clouds with noise and missing data is an ill-conditioned problem, and optimization is the best solution to this problem. Based on the above assumptions, researchers have constructed cell complexes to represent indoor spaces. The cell complex is constructed by taking the disjoint union of zero-dimensional, one-dimensional, two-dimensional and three-dimensional cells, called vertices, edges, polygons and polyhedrons, respectively. The vertexes, segments and facets elements of the divided 3D entity and their spatial topology are uniquely determined [16]. The advantage of the cell complex ensures topological consistency. The indoor 3D modeling is converted into an optimization problem based on the cell complex. According to the different optimization algorithms used, the methods can be divided into two categories: one is integer linear programming, which selects the optimal subset from the candidate facets of the cell complex to form a closed polygonal surface model, but this method ignores the semantic information of the indoor space; the other is graph-based combinatorial optimization, which transforms the indoor modeling problem into the optimal labeling problem of the cell complex, which is expressed by the Markov random field (MRF) model. This method relies on initial room semantic segmentation.

Some scholars have segmented multiple floors and realized model reconstruction by the single-floor method [30]. The 2D method is extended to 3D spatial layout reconstruction without considering the geometric and topological relationships between floors, which cannot meet the reconstruction of cross-floor space. Key features of the proposed method and the state-of-art methods are compared and summarized in Table 1.

Table 1. Overview of recent 3D layout reconstruction methods for indoor environments.

References	Type	Multiroom	Single Floor	Multistory	Cross-Floor	2.5D Assumption		Slanted Surface	Prior Knowledge Adding to Data
						Orthogonal Walls (MW)	Nonorthogonal Walls		
Khoshelham and Diaz-Vilarino [20]	Rule-based	✓	✓	○	○	✓	○	○	○
Becker et al. [21]	Rule-based	✓	✓	✓	○	✓	✓	○	○
Ikehata et al. [22]	Rule-based	✓	✓	○	○	✓	○	○	○
Murali et al. [23]	Data-driven	✓	✓	○	○	✓	○	○	Graph of connected walls
Armeni et al. [17]	Data-driven	✓	✓	○	○	✓	○	○	○
Yang et al. [25]	Data-driven	✓	✓	✓	○	✓	✓	○	○
Mura et al. [26]	Data-driven	✓	✓	✓	○	✓	✓	✓	Every room has at least one scan position
Ambrus et al. [27]	Data-driven	✓	✓	○	○	✓	✓	○	Synthetic sensor poses/viewpoints
Oesau et al. [30]	Data-driven	○	✓	✓	○	✓	○	○	○
Ochmann et al. [31]	Data-driven	✓	✓	✓	○	✓	✓	○	Every room has one scan position
Ochmann et al. [19]	Data-driven	✓	✓	✓	○	✓	✓	○	Oriented point clouds and visibility analysis between patches
Lim and Doh [9]	Data-driven	✓	✓	✓	○	✓	✓	○	Loop close trajectory to form a trajectory cluster
Cui et al. [28]	Data-driven	✓	✓	✓	○	✓	✓	○	Visibility analysis between trajectory and patches
Our method	Data-driven	✓	✓	✓	✓	✓	✓	✓	○

Note: ✓ indicates that the method supports this kind of data; ○ indicates that this kind of data is not supported by the method. The “multistory” mark in Table 1 is based on whether the multistory data experiment is conducted in the article, which does not indicate the ability for multistory modeling. In most cases, multiple floors can usually be divided into single-floor modeling problems.

3. Methods

3.1. Overview

This section introduces our polygonal 3D layout reconstruction method for indoor environments. The inputs include either a point cloud or a mesh. If we define the 3D building model as a space with a boundary surface, then the point cloud or mesh contains two informational aspects: one aspect is the geometric information of sampling points on the object surface, and the other is the spatial information.

Figure 1 depicts the flowchart for the proposed polygonal reconstruction method. The method is divided into five major phases.

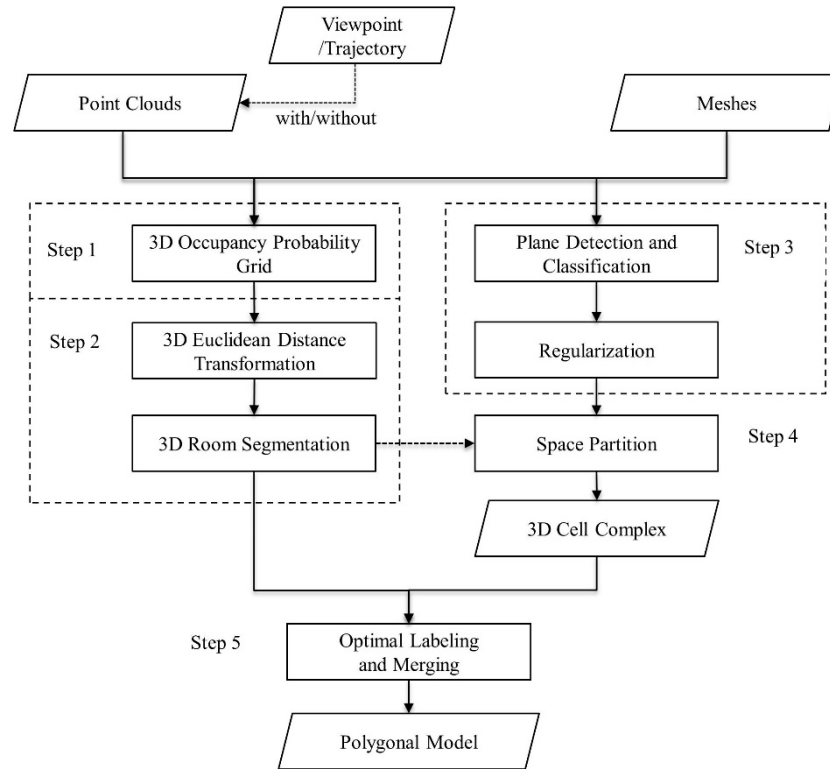


Figure 1. The flowchart of the proposed polygonal reconstruction method.

3.2. 3D Occupancy Probability Grid

Two types of 3D point clouds are utilized as the inputs. One is a point cloud captured by TLS, $P = \{v_\tau, P_\tau\}_{\tau=1}^N$. Each frame of the scanned point cloud is associated with a viewpoint v_τ . The world coordinate system is used to determine the $\{x_i, y_i, z_i\}$ coordinates of point $p_i \in P_\tau$. The other type is a point cloud captured by MLS, $P = \{p_i, \varphi_i\}_{i=1}^N$. Each scanned point is associated with pose information. As shown in Figure 2c, the bounding box is discretized into $m \times n \times h$ grids, and the 3D Bresenham line algorithm (Algorithm 1) is used to calculate the 3D occupancy probability of voxels in the grid [25]. The value of a voxel p is labeled with one of three marks $s_p = \{free = 0, occupied = 1, unknown = -1\}$. A 3D occupancy probability grid map is generated to represent the certainty with which a voxel is occupied by obstacles. The probability value is stored in the 3D grid data structure based on VDB [32], which is an efficient and sparse volume data structure.

Sensors such as monocular cameras and RGB-D sensors are now widely used for interior reconstruction. The output is often a surface mesh that can fulfill the visualization purpose. The mesh datasets are unstructured and contain no semantic information. As these devices are often less expensive than LiDAR, it is important to reconstruct the 3D layout of buildings from these datasets. Multiview geometry and a multiview reconstruction algorithm are used in the reconstruction process, and the pose information is used in the multiview stereo (MVS) to obtain the oriented mesh. Because the viewpoint information

is fused in the dense reconstruction process, the viewpoint information cannot be used directly in a mesh model for occupancy probability calculation. The indoor and outdoor environments can be distinguished by using the normal information of triangles. The value of a voxel p is labeled with one of three marks $s_p = \{free = 0, occupied = 1, unknown = -1\}$. The algorithm contains three main steps (Algorithm 2). First, all the grid voxels are initialized with free values. Second, the voxel located on the triangle is marked with an occupied value. The normal vector is classified into 26 regular directions, and the encoded direction value of the voxel ranges from 1 to 26 (the initial direction value is 0). Third, a wavefront algorithm checks the 26 adjacent voxels around the current voxel and updating the direction of the current voxel. If all 26 adjacent voxels' directions point toward the current voxel, the direction value of the current voxel after wavefront growth is set to 27. If the direction value of a voxel equals zero, the corresponding voxel in the occupancy probability grid is set to unknown. If the direction value of a voxel is greater than zero, the corresponding voxel in the occupancy probability grid is set to free.

Algorithm 1. 3D occupancy probability calculation for a point cloud.

Input:

P_τ // input point cloud

v_i // viewpoint

$bbMax, bbMin$ // bounding box of the point cloud

Initialize:

$M_{occ} \leftarrow \emptyset$ // 3D occupancy probability grid

(1) for each ($p_i \in P_\tau$)

(2) // calculate the coordinates of occupied and free voxels in the grid

(3) bresenham_in_3D(v_i, p_i, M_{occ});

(4) end for

(5) return M_{occ} ;

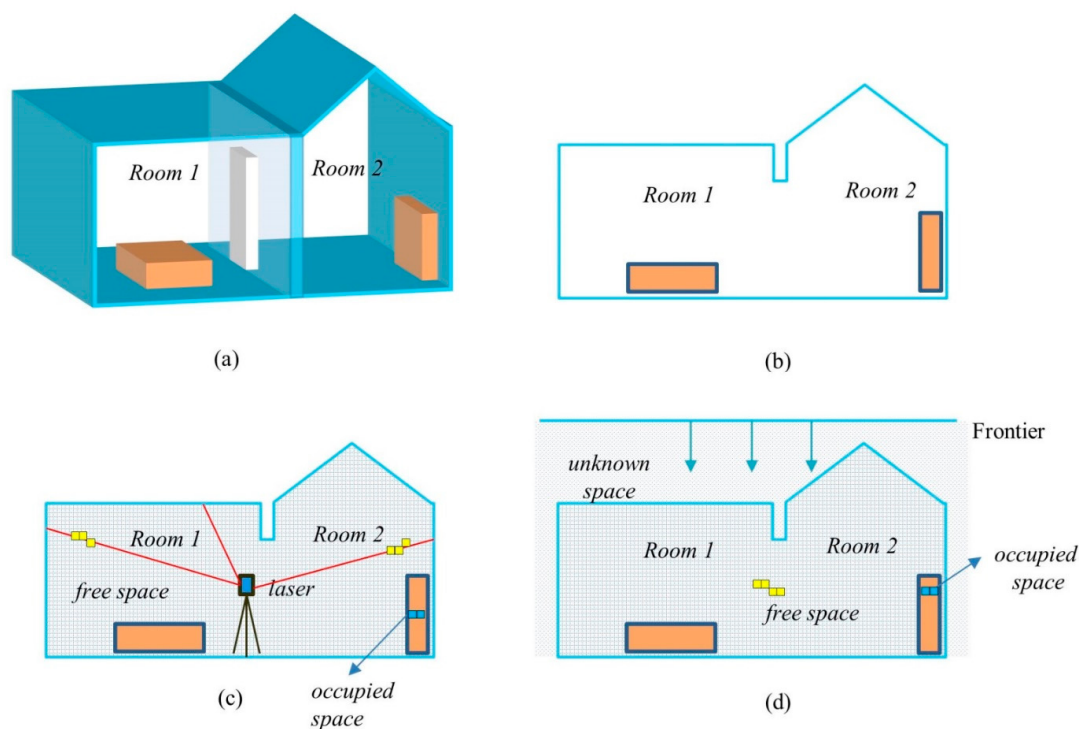


Figure 2. Example of voxelization of an indoor space: (a) door separating two rooms; (b) profile of the indoor space; (c) point cloud acquisition using laser scan, where the voxels are marked as free spaces and occupied spaces are colored yellow and blue; (d) voxelization for a mesh without poses/viewpoints, frontier is used to mark unknown voxels.

Algorithm 2. 3D occupancy probability calculation for an oriented mesh.

```

Input:
 $P_{mesh}(P, T)$ ; // normal oriented mesh
 $bbMax, bbMin$ ; // bounding box of the point cloud
Initialize:
 $M_{occ} \leftarrow \emptyset$ ; // 3D occupancy probability grid
 $M_{dir} \leftarrow \emptyset$ ; // D-26 direction grid
(1) for each ( $t_i \in T$ )
(2)   box = getBoundingBox( $t_i$ ); // calculate the coordinates of occupied and free points in the
metric map
(3)   for i = minx:maxx
(4)     for j = miny:maxy
(5)       for k = minz:maxz
(6)          $s_p = \text{voxel}(i, j, k)$ ;
(7)         if ( $s_p.\text{inside}(t_i)$ )
(8)            $M_{occ}.\text{setValue}(s_p, 255)$ ;
(9)            $N = \text{getNormal}(t_i)$ ;
(10)           $D = \text{normalTo26Directoin}(N)$ ;
(11)           $M_{dir}.\text{set}(s_p, D)$ ;
(12)        end if
(13)      end for
(14)    end for
(15)  end for
(16) end for
(17)  $M_{dir} = \text{wavefront}(M_{dir})$ ;
(18)  $M_{occ} = \text{updateUnknown}(M_{occ}, M_{dir})$ ;
(19) return  $M_{occ}$ ;

```

However, in practical application, the existence of point cloud holes affects the indoor and outdoor differentiation effects of the directional mesh. At the same time, due to the existence of indoor green plants and other objects, the normal vectors in some areas are messy and cannot meet the need to distinguish between free space and unknown space. Therefore, a more general algorithm is proposed that assumes that the mesh model meets the vertical orientation and the ceiling area is completely scanned, which is easy to satisfy for most data-acquiring devices (Algorithm 3).

The first step is to initialize the occupancy probability grid and set the value of each voxel to free. Then, the voxel located on the triangle is marked with an occupied value. The last step marks the unknown area. First, we initialize the frontier (the maximum value of the point cloud elevation range), set each voxel as unknown and traverse the voxel layer by layer according to the z-axis direction (Figure 2d). When the voxel value of the current layer in the occupied grid is the same as that of the previous layer, this value is marked as unknown, and if the voxel value of the current layer is different from that of the previous layer, traversal ceases. Finally, all unknown areas are marked. The returned occupancy probability grid contains $s_p = \{free = 0, occupied = 1, unknown = -1\}$ three values.

3.3. 3D Room Segmentation

A significant difference between indoor space and outdoor space is the cellular nature of indoor space. Taking the fact that local maxima of distance transform (DT) values usually reside in the middle of a room and that rooms are connected by small passageways (doors and junctions), the 3D room segmentation method [33] utilizes the volumetric representation and sphere packing of indoor space to separate rooms as connected components. This method contains five main steps. First, the method applies 3D Euclidean distance transformation (EDT) to the occupancy probability grid to determine how far each voxel is from the closest occupancy point. Second, the distance map acquired following the EDT is then divided into segments according to the specified distance threshold. Third, areas with distance values larger than the specified threshold are filled with inner spheres. A

topological graph is constructed according to the adjacency relationship between the filled inner spheres. Fourth, the connected subgraphs of the topological graph are segmented (Figure 3). By adding the space occupied by each inner ball to the connected subgraph, the initial room seed region is obtained. Fifth, the wavefront growth algorithm starts with a seed voxel in each seed room area and uses the wavefront growth algorithm to determine the unlabeled voxels belonging to the same room to obtain the final 3D room segmentation result. Free space is labeled with room information after 3D room segmentation (Figure 4). In this study, to speed up the filling speed of the inner ball, we used a combination of random filling and ordered filling.

Algorithm 3. 3D occupancy probability calculation for a general mesh.

```

Input:
 $P_{mesh}(P, T)$  // general mesh
 $bbMax, bbMin$  // bounding box of the mesh
Initialize:
 $M_{occ} \leftarrow \emptyset$ ; // 3D occupancy probability grid
(1) setValue( $M_{occ}$ , FREE);
(2) for each ( $t_i \in T$ )
(3)   box = getBoundingBox( $t_i$ ); // calculate the coordinates of occupied and free points in the
metric map
(4)   for i = minx:maxx
(5)     for j = miny:maxy
(6)       for k = minz:maxz
(7)          $s_p = \text{voxel}(i, j, k)$ ;
(8)         if ( $s_p.\text{inside}(t_i)$ )
(9)            $M_{occ}.\text{setValue}(s_p, \text{OCCUPIED})$ ;
(10)        end if
(11)      end for
(12)    end for
(13)  end for
(14) end for
(15) updateUnknown( $M_{occ}$ );
(16) return  $M_{occ}$ ;

```

The function of room segmentation can be divided into two aspects: one is to classify detected planes from point cloud or mesh according to each room to reduce the complexity of space partition. The other is to generate semantic information and provide semantic initial values for room layout reconstruction.

3.4. Plane Detection and Classification

The global L0 gradient minimization-based plane detection method [34] is adopted for the point cloud. It is a fast algorithm for regularity-constrained plane fitting (RCPF) problems. Region growing is used for plane detection from meshes (Figure 5). Due to the inevitable error of the measuring equipment, the extracted planes are not continuous to build watertight polygonal models. A semantic enhancement and rejected strategy are employed. For room layout reconstruction, only wall, ceiling and floor planes are considered in the next space partition step.

The deviation of unit normal n_i from plane p_i with regard to vertical axis $n_z = (0, 0, 1)^T$ is measured by vertical attribute $a_h = |n_i \cdot n_z|$. The plane is categorized as a wall if $a_h < \epsilon$. The threshold ϵ , for example, $\epsilon = \cos(90^\circ \pm 10^\circ)$, used to identify a vertical plane is the variable, which is the angle threshold's cosine value. Walls with a height difference of less than 1.5 m are not considered. Additionally, small plane patches, including flat and slant planes with an area of less than 1 m², are also removed. Because the automatic plane extraction method cannot be guaranteed to be completely correct, it is necessary to correct the errors interactively.

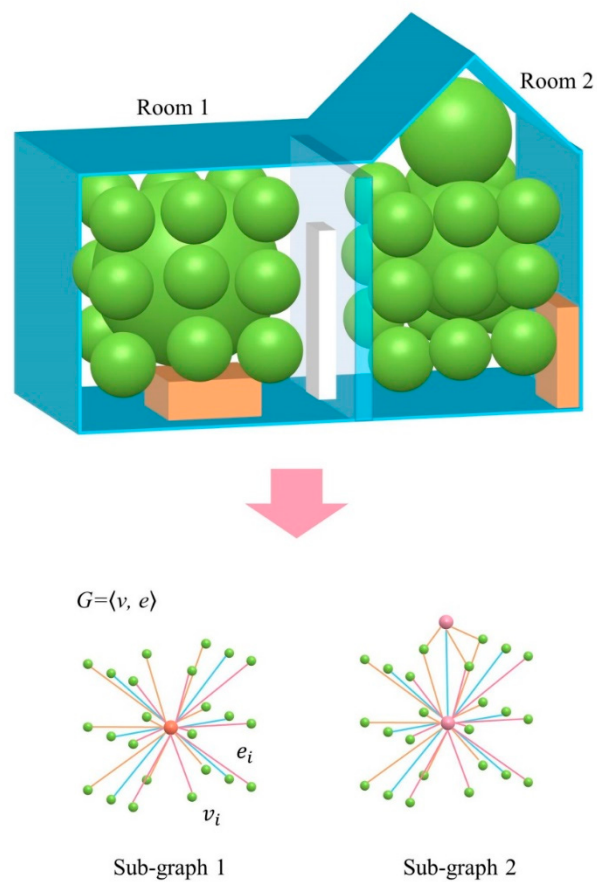


Figure 3. Inner sphere packing of the indoor space and constructed graph.

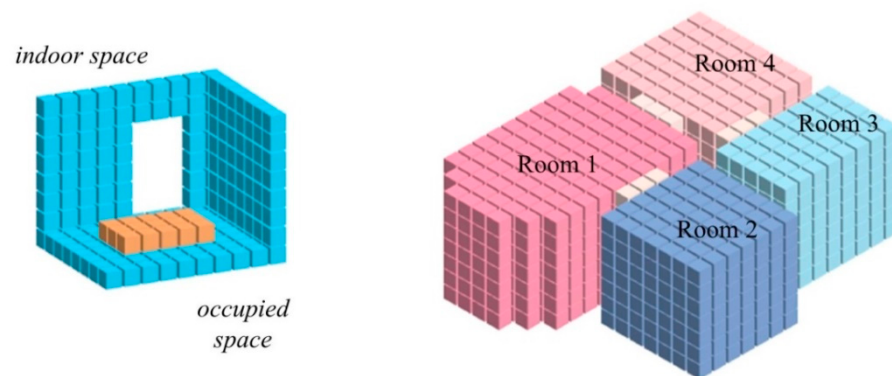


Figure 4. Occupied space and free space is labeled with room information after 3D room segmentation.

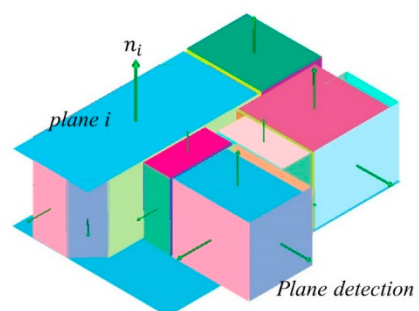


Figure 5. Detected planes from the mesh and colored in different colors.

3.5. Regularization

Because there are many extracted planes, too many planes with slight difference will make the space partition very complex and increase the computational burden. The angles of the plane normals and the distances between the planes are often used to compute the difference. Therefore, we cluster the approximate planes through the regularization process. The regularization process includes vertical orientation and normal vector clustering.

First, we orient the planar walls to make the walls orthogonal to the floor. The constrained least square method is used to realize the plane refitting of the wall point cloud. Second, we construct the nearest neighborhood M_i for each plane i . If the normal n_i of plane i and n_j of plane j satisfy $|n_i \cdot n_j| \geq 1 - \cos\beta$, plane j is regarded as a neighbor of plane i and added to the neighborhood M_i . Finally, parallel plane clusters are created using Gaussian-map clustering. Each plane is given a weight equal to the number of related points and is projected onto the unit sphere by its normal. By taking into account the mirrored point on the sphere, the normal of the cluster is also oriented. The mean-shift algorithm [35] is used to extract the peaks. All planes within one peak are considered to be parallel, and the normal of the cluster is defined as the average normal of parallel planes, weighted with the area of each plane. The adjusted planes in the cluster are used for space partitioning.

3.6. Space Partition

In this step, we take a set of planes as input and return a partition of the bounding 3D space. The goal of space partition is to generate cell complex, and we adopt a relatively simple implementation. The method utilizes a polygon mesh data structure [36] to store the subdivided space. The detected planes are classified for each room based on the room segmentation results. The space partitioning is performed per room. A binary space partition (BSP) is used to maintain the hierarchical relationship of the subdivided space. A polygon mesh is a group of vertices, edges, and faces that determines how a polyhedral object is shaped. A polygon mesh is a half-edge data structure that is oriented consistently in counterclockwise order around each face and along each boundary. It is considered to have the topology of a 2-manifold. The basic idea of binary space partitioning is that any plane can divide the space into two half spaces. For a bounding 3D space or subspaces, all points on one side of the split plane define a half space, and points on the other side of the split plane define another half space. If we continue to define a plane in any half space, we further divide the half space into two smaller subspaces. Continuing this process, the subspace is segmented to be progressively smaller, and finally, a spatial binary tree is formed. Each subspace constructs a convex polyhedron. The topological geometric relationship between different polyhedrons can be effectively expressed by searching the BSP tree (Figure 6). The space partition algorithm is outlined in Algorithm 4.

3.7. Optimal Labeling and Merging

After space partitioning, the 3D space is partitioned into a set of polyhedron cells. The polyhedron cells with the same label are considered in the same room. An undirected graph $G = \langle v, e \rangle$ is defined to encode the set of polyhedron cells to the label set L . Node v represents the polyhedron cell in the cell complex, and edge e represents the topological relationship between cells. The neighbors of node v are stored in N . The energy function for the labeling problem is defined as:

$$U(l) = \sum_{i \in v} D_i(l_i) + \sum_{ij \in N} V_{ij}(l_i, l_j) \cdot T(l_i \neq l_j) \quad (1)$$

Unary energy

The unary energy is designed to describe the likelihood that a polyhedron belongs to a label. The value is computed as the ratio of the volume of occupied voxels with label l_i to

the volume of subdivided polyhedron. Whether a voxel is inside a polyhedron is tested using the function of the polygon mesh.

$$D_{i \in v}(l_i) = -\ln \left(\frac{\text{count}(\text{voxel} = l_i \text{ and } \text{voxel} \in \text{cell}) * s_{\text{voxel}}^3}{\text{volum}(\text{cell})} \right) \quad (2)$$

Algorithm 4. Space partition.

Input:

S_{plane} // input planes

$bbMax, bbMin$ // 3D bounding box of the point cloud

Initialize:

Polygon _mesh vol; // a polygon mesh

BSP_tree bsp;

(1) vol = buildBounding3DSpace($bbMax, bbMin$);

(2) bsp.AddNode(vol);

(3) for each ($s_i \in S_{\text{plane}}$)

(4) vol_tmp = bsp.getBeginNode();

(5) while(vol_tmp)

(6) [vol1, vol2] = clip($s_i, \text{vol_tmp}$);

(7) bsp.update(vol_tmp, vol1, vol2);

(8) vol_tmp = bsp.getNextNode();

(9) end while

(10) bsp.getBeginNode();

(11) end for

(12) return bsp;

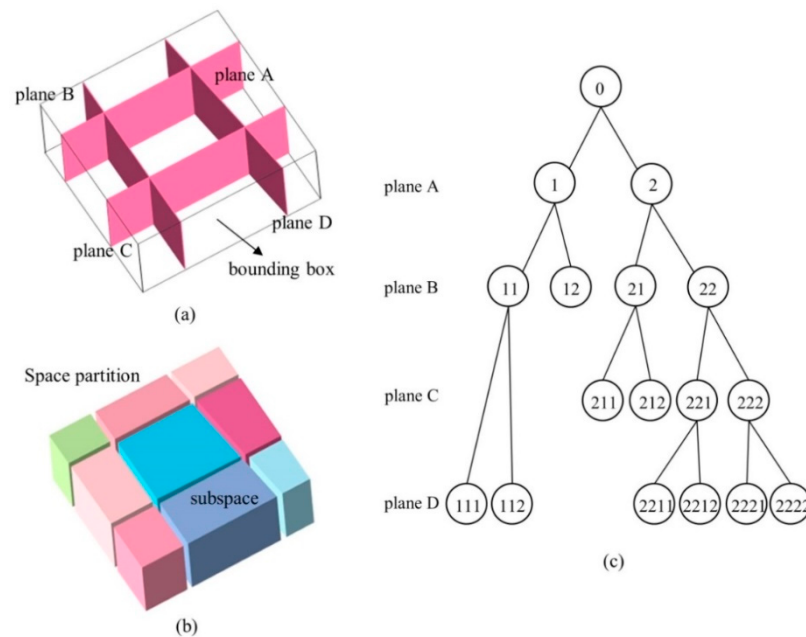


Figure 6. (a) The bounding 3D space is partitioned with four planes; (b) the bounding 3D space is divided into nine subspaces which correspond to the leaf node of BSP tree; (c) a plane divides the space into two half spaces and BSP tree is used to maintain the hierarchical relationship of the subdivided space.

Pairwise energy

The pairwise energy describes the likelihood that two polyhedron cells are contained within the same room. It is defined as the ratio between voxels occupied by the original wall patch and voxels in the cell complex facet. A weaker relationship between two polyhedron

cells is indicated by a higher likelihood between them. A wall separating two cells indicates that they are unlikely to be in the same room.

$$V_{i,j \in N}(l_i, l_j) = \begin{cases} -\ln\left(1 - \frac{\text{count}(\text{voxel} = \text{occupied and voxel} \in \text{facet}_{\text{cell}})}{\text{count}(\text{voxel} \in \text{facet}_{\text{cell}})}\right), & \text{if } l_i \neq l_j \\ 0, & \text{if } l_i = l_j \end{cases} \quad (3)$$

The graph cut approach is applied to minimize the objective function to obtain optimum labeling. The room layout reconstruction is transformed into an energy minimum and can be solved through the α - β swap algorithm [37]. Then, rooms are created by merging all the polyhedron cells with the same label.

4. Results

The program language is C++. The Computational Geometry Algorithms Library (CGAL) [38] and Cloud Compare [39] are used in the algorithm's implementation. All experiments are performed with an Intel Core i7-10750H CPU (2.60 Hz) and 16 GB of RAM. The quantitative evaluation method is used to evaluate the reconstruction results [40,41]. By comparing the reference model R with the reconstruction polygon model S called the source model, the completeness, correctness and accuracy are quantitatively evaluated. The definition of completeness is:

$$M_{\text{Comp}} = \frac{\sum_{i=1}^n \sum_{j=1}^m |S^i \cap b(R^j)|}{\sum_{j=1}^m |R^j|} \quad (4)$$

Among them, the intersection area is calculated on all facets between S^i and R^j , and the completeness changes with the size of buffer b . The definition of correctness is:

$$M_{\text{Corr}} = \frac{\sum_{i=1}^n \sum_{j=1}^m |S^i \cap b(R^j)|}{\sum_{i=1}^n |S^i|} \quad (5)$$

Accuracy is defined as:

$$M_{\text{Acc}} = \text{Med} \|\pi_j^T p_i\|, \text{ if } \|\pi_j^T p_i\| \leq r \quad (6)$$

where $\|\pi_j^T p_i\|$ is the vertical distance between vertex p_i in the source model and plane π in the reference model and r is the truncation threshold to avoid the influence of an incomplete or inaccurate source model. Relatively high completeness and low correctness scores mean that the reconstructed models contain most of the elements that are present in the corresponding reference models but that they also include a considerable number of incorrect facets.

4.1. Point Cloud Datasets

We first test our method on four point cloud datasets, named A1, A2, A3 and A4. A1, A2 and A3 are derived from the University of Zurich (UZH) Rooms detection datasets [26]. They are scanned with a Faro Focus 3D laser range scanner. The A4 dataset is derived from the International Society for Photogrammetry and Remote Sensing (ISPRS) Benchmark on Indoor Modeling [42]. The datasets are captured by MLS (i.e., Zeb-Revo). This dataset consists of point clouds, corresponding trajectory information and timestamps. The dataset is preprocessed by aligning the coordinate points and trajectories with timestamps. The point cloud noise level of A4 is low, and the relative accuracy is 2–3 cm.

The original point clouds are shown in Figure 7, and the descriptions of the dataset are listed in Table 2. The room segmentation results for each dataset are shown in the second column in the figure with different colors. The layout of the datasets is reconstructed and

represented as polygon models in the third column. The input parameters for different datasets are listed in Table 3.

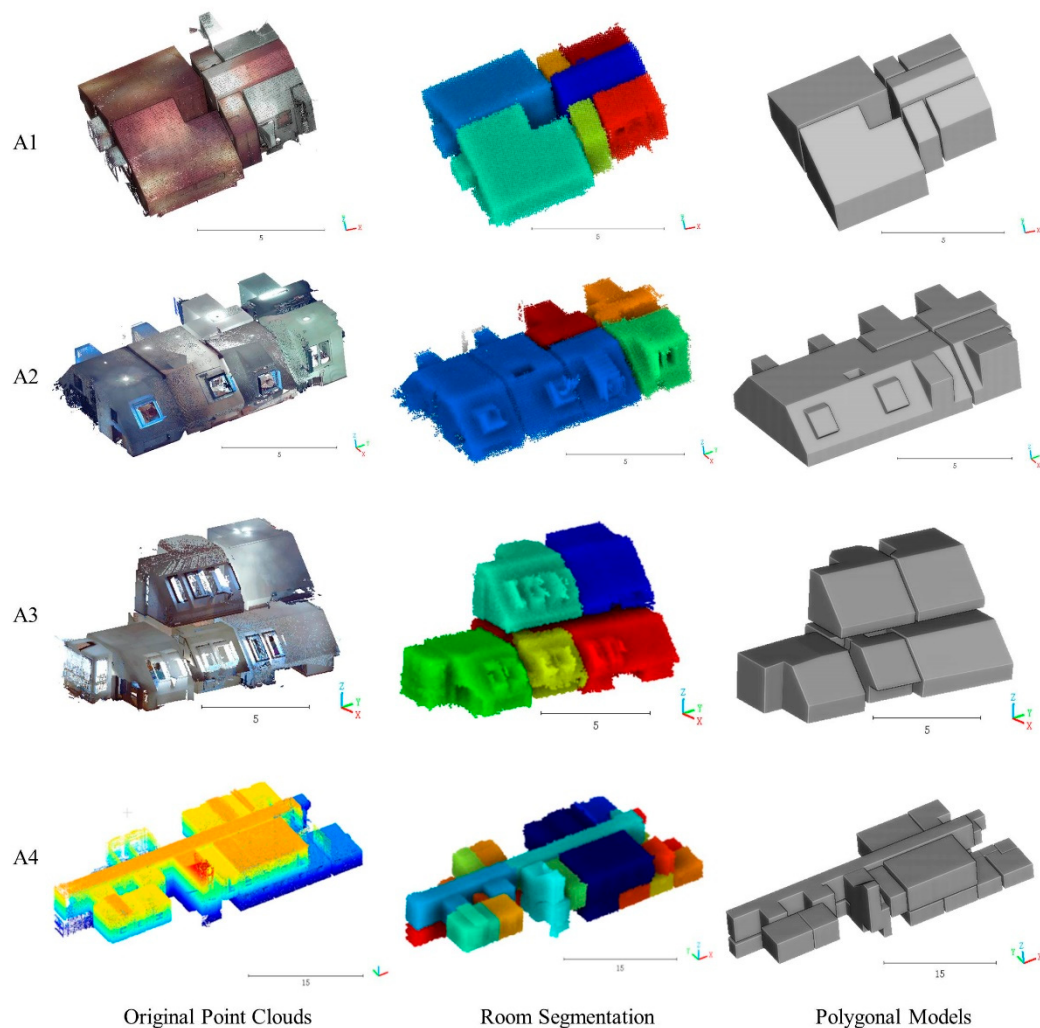


Figure 7. Experiments on point cloud datasets. From left to right: Original point cloud, room segmentation result, reconstructed room layout.

Table 2. Detailed descriptions of the experimental point cloud datasets.

Test Sites	Frames	Size (m)	Points	Single Floor	Multistory	2.5D Assumption		Slanted Walls/Sloped Ceilings
						Orthogonal Walls (MW)	Nonorthogonal Walls	
A1	7	$9.0 \times 8.0 \times 3.2$	12.4×10^6	✓	○	✓	○	✓
A2	7	$7.9 \times 12.0 \times 2.8$	19.0×10^6	✓	○	✓	○	✓
A3	8	$7.7 \times 13.8 \times 6.2$	21.9×10^6	○	✓	✓	✓	✓
A4	-	$41.8 \times 16.5 \times 8.5$	21.8×10^6	○	✓	✓	○	○

Note: ✓ indicates that the data has the characteristics of this aspect; ○ indicates that the data doesn't have the characteristics of this aspect.

Each reconstructed model's completeness, correctness and accuracy are calculated for buffer sizes and cutoff distances ranging from 1 cm to 15 cm. For completeness and correctness, larger numbers suggest more complete and correct models, whereas smaller values for accuracy indicate the better accuracy of the reconstructed models. The execution time includes three parts: room semantic segmentation, plane segmentation and optimization modeling.

Table 3. Input parameters for the point cloud datasets.

Parameters	Descriptions			
3D room segmentation				
s_{voxel}	Voxel size of the grid map.			
Δd	Distance threshold for distance map segmentation.			
$\delta_{overlap}$	Overlap ratio between two spheres			
τ	Minimum volume of a seed room region			
Plane detection				
k	Searching k nearest neighbors			
ns	Minimum support points of a plane			
np	Number of plane normals			
Parameters	A1	A2	A3	A4
s_{voxel}	0.08 m	0.08 m	0.08 m	0.1 m
Δd	0.7 m	0.7 m	0.8 m	1.0 m
$\delta_{overlap}$	0.8	0.8	0.8	0.8
τ	0.03 m ³	0.3 m ³	0.3 m ³	0.3 m ³
k	10	10	10	10
ns	50	50	50	50
np	30	30	30	30

The A1 dataset contains point clouds of single-floor building interiors with orthogonal and slant walls. As shown in Figure 7, seven rooms are detected. The results show that the reconstruction methods achieve the highest completeness and correctness. The accuracy is 0.56 cm at a buffer size of 10 cm. A2 contains point clouds of single-floor building interiors with orthogonal and slant walls. As shown in the figure, four rooms are detected. A2 achieves high completeness of 99.3% and correctness of 99.0% at a buffer size of 10 cm. The execution time of optimization modeling is the longest of all datasets because the room is quite complex.

A3 contains point clouds of large-scale building interiors with nonorthogonal walls. Seven rooms are reconstructed. The A3 dataset reached 83.6% completeness and 81.3% correctness when using a 10 cm buffer. The experiments on A1, A2 and A3 prove the effectiveness of 3D room segmentation for multistory indoor environments that contain wall structures with arbitrary orientations. A4 represents a double-story office building. The point clouds contain orthogonal walls, and two floors are connected by stair. A4 achieves a completeness of 90.2% and a correctness of 92.9% at a buffer size of 10 cm but with the lowest completeness and correctness values at small buffer sizes. The accuracy is 3.0 cm, which is the highest and indicates the lowest accuracy.

Figure 8 compares the completeness, correctness and accuracy of polygon models to the reference model. The models built for dataset A1 are generally more accurate and complete than those for other datasets, according to a comparison of the results. This can be attributed to the dataset's lower complexity and, possibly, higher data quality. The lowest accuracy of A4 is attributed to the worse data quality. The results presented in the table demonstrate that the proposed method successfully models the entire dataset (Table 4).

The algorithm counts only the execution time of the program, including room segmentation, plane detection and optimization modeling, without considering the time of human interaction. The more complex the structure of the room, namely, the more planes used for space partition, the longer the execution time of the program.

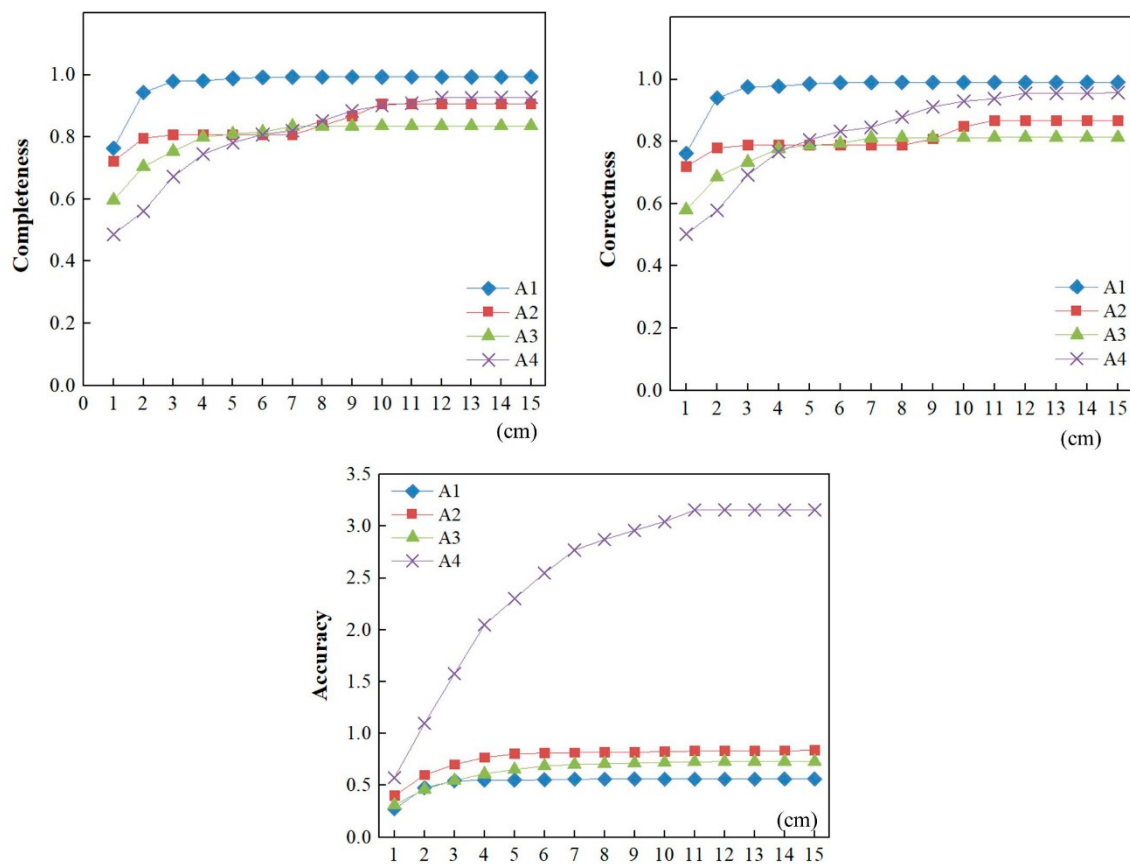


Figure 8. Completeness, correctness, and accuracy of each reconstructed polygon model.

Table 4. The evaluation results for point cloud datasets.

Test Sites	Detected Rooms	Completeness @ 10 cm	Correctness @ 10 cm	Accuracy @ 10 cm (cm)	Time (s)		
					Room Segmentation	Plane Detection	Optimization Modeling
A1	7	99.3%	99.0%	0.56	4.561	21.477	44.055
A2	4	90.7%	84.8%	0.83	8.331	29.655	365.782
A3	6	83.6%	81.3%	0.72	16.756	23.339	333.616
A4	20	90.2%	92.9%	3.0	461.625	16.69	202.027

4.2. Mesh Datasets

In order to confirm the method's efficacy on meshes, four mesh datasets of interior scenes from the Matterport3D dataset [41] and Stanford 2D–3D-semantic dataset [17] were used to test the proposed method. The datasets were captured by RGB-D cameras in a variety of indoor buildings. The parameter descriptions are listed in Table 5.

The 3D occupancy probability grid's voxel size is set to the same value of 0.08 m, as shown in Table 6. The lower bound of the room area is the same. The original meshes, room segmentation results and reconstructed layout of each dataset are displayed in Figure 9. With a buffer size of 10 cm and a cutoff distance ranging from 1 to 15 cm, Figure 10 compares the completeness, correctness and accuracy of polygon models with the reference model.

The B1 dataset contains meshes of single-floor building interiors with nonorthogonal walls. Six rooms are detected and reconstructed. As shown in Table 7, B1 achieves 85.1% completeness and 91.6% correctness at a buffer size of 10 cm. The dataset represents relatively simple environments.

Table 5. Detail descriptions of the experimental mesh datasets.

Test Sites	Size (m)	Points	Triangles	Single Floor	Multi-Storey	2.5D Assumption		Slanted Walls/ Sloped Ceilings
						Orthogonal Walls (MW)	Non-Orthogonal Walls	
B1	11.8 × 9.2 × 3.5	166,336	648,654	✓	○	✓	✓	✓
B2	15.0 × 8.4 × 3.1	124,032	408,782	✓	○	✓	✓	✓
B3	29.7 × 25.8 × 11.4	159,284	139,577	✓	○	✓	✓	○
B4	22.4 × 15.3 × 11.0	201,328	753,526	○	✓	✓	○	✓

Note: ✓ indicates that the data has the characteristics of this aspect; ○ indicates that the data doesn't have the characteristics of this aspect.

Table 6. Parameters and descriptions of the proposed method for mesh datasets.

Parameters	Descriptions			
3D room segmentation				
s_{voxel}	The voxel size of grid map.			
Δd	Distance threshold for distance map segmentation.			
δ_{overlap}	The overlap ratio between two sphere			
τ	Minimum volume of a seed room region			
Region growing plane detection				
Δd_2	Distance threshold between point to the plane			
θ	Angle threshold for normals between vertex and plane			
t_{area}	Small area planes are rejected			
Parameters	B1	B2	B3	B4
s_{voxel}	0.08 m	0.08 m	0.08 m	0.1 m
Δd	0.7 m	0.7 m	0.8 m	1.0 m
δ_{overlap}	0.8	0.8	0.8	0.8
τ	0.03 m ³	0.3 m ³	0.3 m ³	0.3 m ³
Δd_2	0.02 m	0.02 m	0.02 m	0.02 m
θ	25°	25°	25°	25°
t_{area}	0.5 m ²	0.5 m ²	0.5 m ²	0.5 m ²

Table 7. Evaluation results for mesh datasets.

Test Sites	Detected Rooms	Completeness @ 10 cm	Correctness @ 10 cm	Accuracy @ 10 cm (cm)	Time (s)		
					Room Segmentation	Plane Detection	Optimization Modeling
B1	6	85.1%	91.6%	0.53	26.732	5.047	33.441
B2	9	92.2%	94.6%	1.5	13.867	2.572	68.211
B3	18	86.8%	86.0%	0.85	72.594	1.435	296.121
B4	20	88.1%	87.2%	1.1	75.636	4.661	258.346

B2 contains meshes of single-floor building interiors with nonorthogonal walls and slanted ceilings. Eight rooms are detected and reconstructed. The findings indicate that the reconstruction methods perform worse when the buffer size is small but quickly improve as the buffer size is increased to 10 cm, where they perform better, with completeness and correctness reaching 92.2% and 94.6%, respectively. Despite the lowest accuracy of all mesh datasets at a buffer size of 10 cm, the accuracy is 1.5 cm (Table 7). This is explained by the fact that of these four datasets, dataset B2 is the least complex and may also have the worst data quality.

B3 contains meshes of single-floor building interiors with nonorthogonal walls, which represents a more complex environment on a large scale. Due to the complexity of the dataset, oversegmentation occurs. The stair region was not segmented out because the original data was very incomplete. The segmented room is further divided using a virtual

closure plane. B3 achieves a completeness of 86.8% and a correctness of 86.0% at a buffer size of 10 cm, which is the lowest of all four datasets. The accuracy is 0.85 cm with a cutoff distance of 10 cm (Table 7).

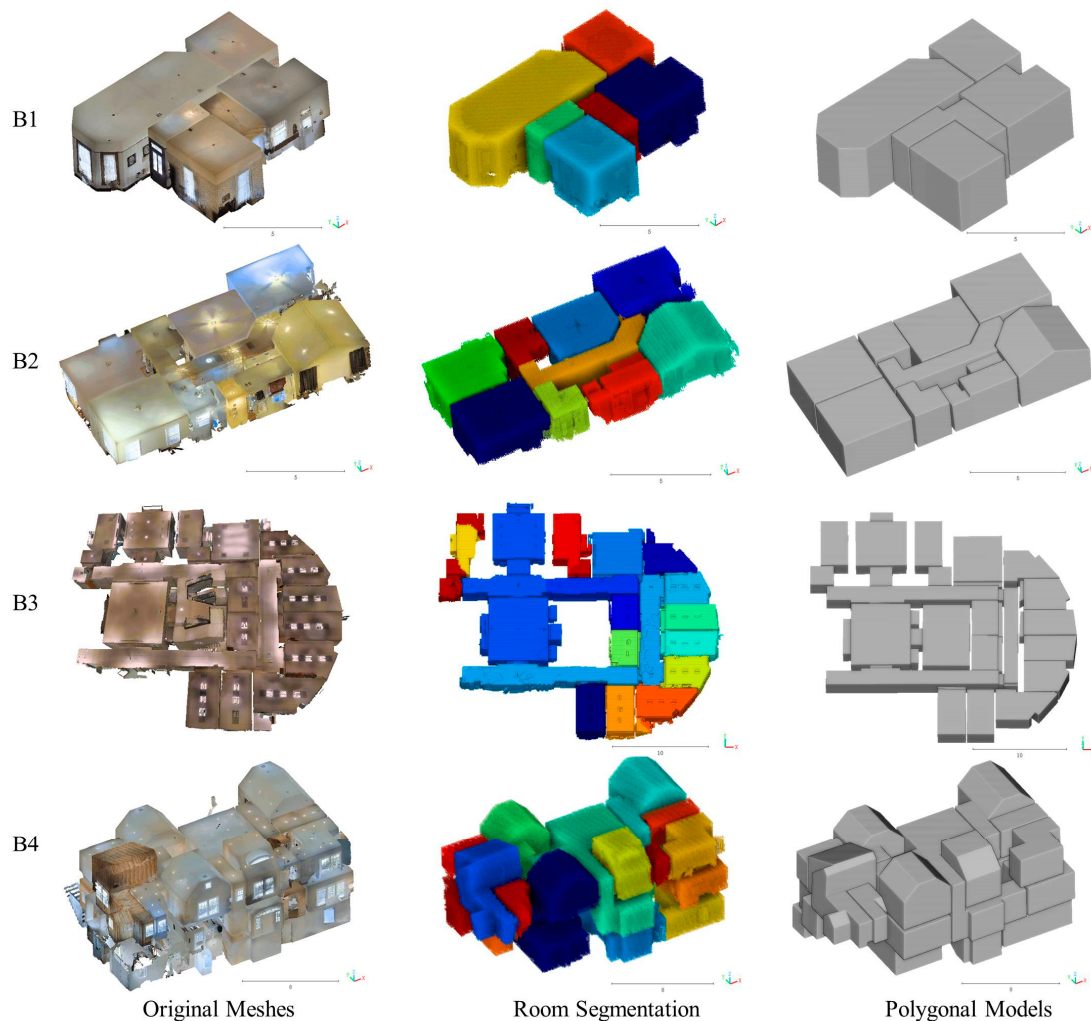


Figure 9. Experiments on mesh datasets. From left to right: Original point cloud, room segmentation result, reconstructed room layout for different meshes.

B4 contains meshes of multistory building interiors. The studied building contains three floors, and the three floors are connected by a staircase and hall. Twenty rooms are detected. As shown in Table 7, B4 achieves a completeness of 88.1% and a correctness of 87.2% when the buffer size is 10 cm. The accuracy is 1.1 cm with a cutoff distance of 10 cm.

The overall accuracy of mesh modeling is high, which is closely related to the high accuracy of the extraction planes for mesh datasets. The reference model is created manually, based on the mesh slices, and hence may differ from the ground truth in some ways.

4.3. Cross-Floor Spaces

The proposed method was also tested on datasets of cross-floor indoor spaces. Figure 11 shows the original point cloud data (Figure 11a), room segmentation results (Figure 11b) and polygonal modeling results. Because the three corridors are connected with the stair space, this space is further divided into three corridors and a stair space using a virtual door plane (Figure 11c). The building space is divided into a 3D cell complex using the segmented planes and semantic labeling. Finally, the 3D polygonal model is obtained (Figure 11e).

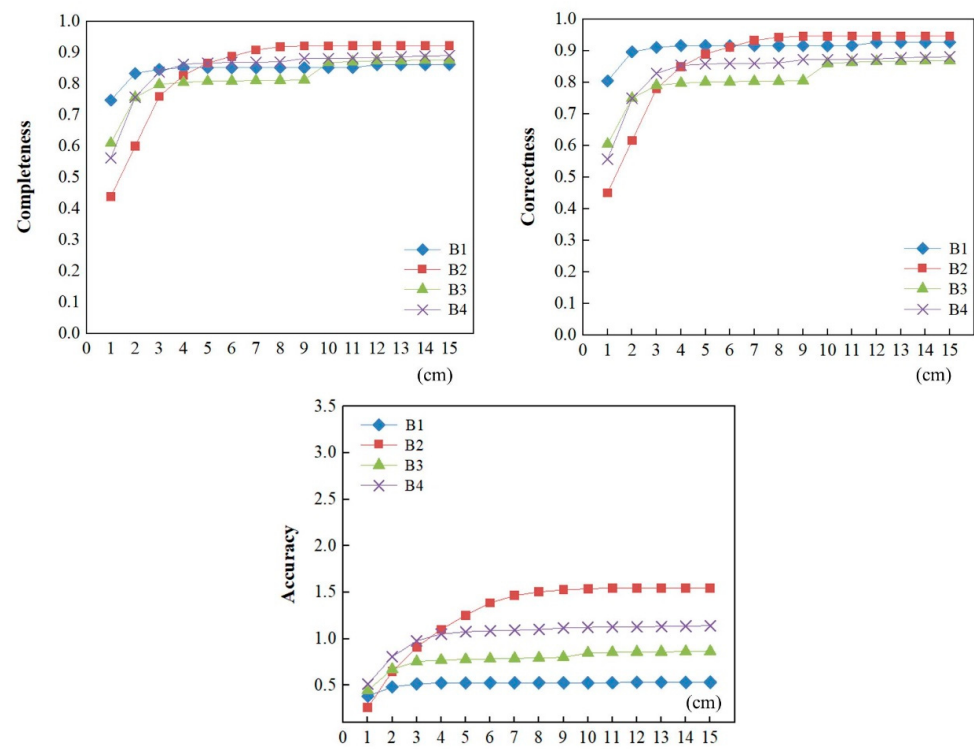


Figure 10. Evaluation results for the reconstructed polygon model on completeness, correctness and accuracy.

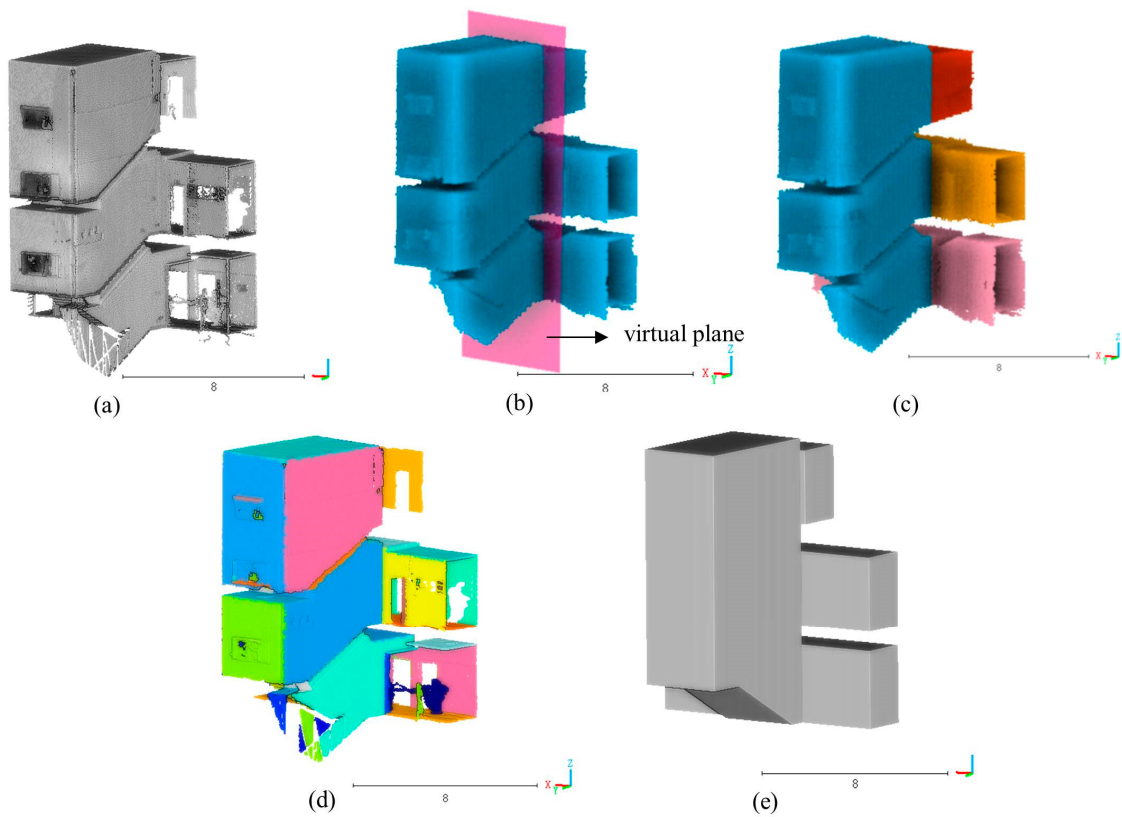


Figure 11. (a) Original point cloud; (b) initial room segmentation result; (c) further division into three corridors and a stair space using a virtual closure plane; (d) segmented planes; (e) reconstructed room layout of cross-floor spaces.

When there are too many planes involved in dividing the bounding 3D space, the process produces a small piece of subspaces, resulting in defects in the result model, although the model is still a watertight polyhedron (Figure 12).

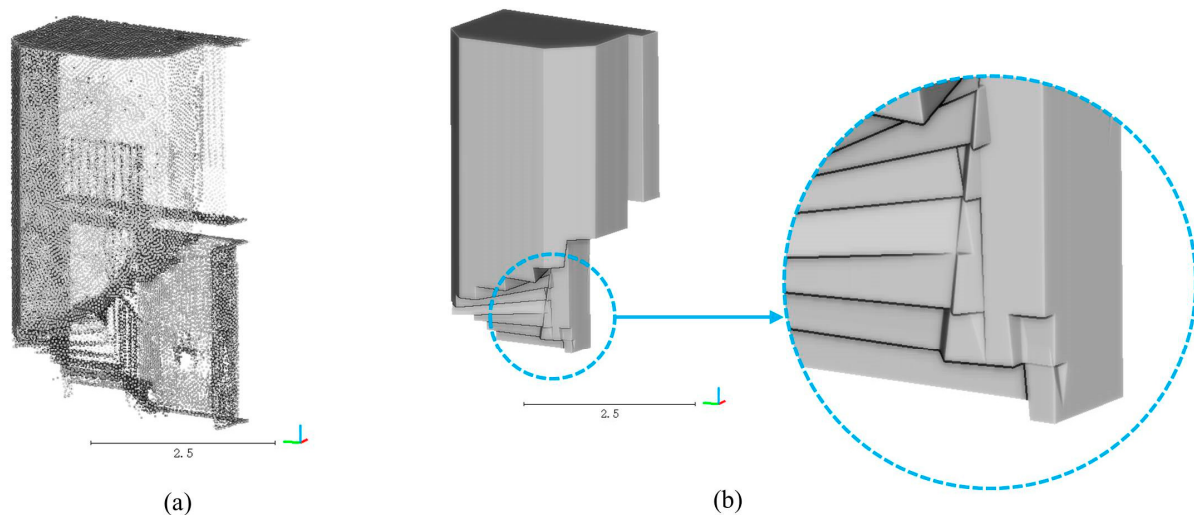


Figure 12. (a) Original point cloud; (b) reconstructed room layout of cross-floor space, where too many small pieces of subspaces result in defects in the result model.

5. Discussion

In this research, both point clouds and meshes are used for room layout reconstruction. When compared with point clouds, meshes have the following advantages: In the process of constructing a mesh through image or image and point cloud fusion, the surface is optimized through the fusion process, and the accuracy of the surface is better than that of raw point clouds. Moreover, the fusion process takes into account the pose/viewpoint information, and the neighbor of each vertex is clear, so parallel plane extraction from meshes is not a problem.

In the modeling process, the method in this study tends to build the connected area into a room. When the opening is large enough to be treated as a virtual door, the semantic segmentation of the room produces undersegmentation, and the room reconstruction process reconstructs the two rooms connected by the virtual door into a whole. Thus, further space partitioning is needed using the virtual closure plane where the virtual door is located. When compared with the existing methods, it can be found that on dataset B4, this method has high completeness, correctness and accuracy.

6. Conclusions

This paper presents a general room layout reconstruction method for building interiors. The method combines voxel-based room segmentation and space partitioning and creates optimal indoor models with room layout information. Various data sources, including point clouds and meshes, can be used as input. The method can be used to reconstruct the layouts of complex indoor environments, including multiple rooms, multiple stories, slanted planes, cross-floor spaces and so on. It provides a general solution for 3D layout reconstruction in a non-Manhattan world.

However, the method is not fully automatic. At present, room segmentation and plane detection are still not completely accurate, and some human interaction is needed. The efficiency of the 3D room segmentation and space partition algorithm needs to be improved. Planned future work includes the development of a more efficient 3D room segmentation algorithm and space partition algorithm.

Author Contributions: Conceptualization, Fan Yang; Data curation, Mingliang Che and Shihua Wang; Formal analysis, Yingli Wang; Funding acquisition, Fan Yang and Chi Zhang; Investigation, Shihua Wang and Jiyi Zhang; Methodology, Fan Yang, You Li and Mingliang Che; Project administration, Chi Zhang; Resources, Shihua Wang; Software, Fan Yang and Xinliang Cao; Supervision, Yingli Wang; Validation, Yingli Wang, Jiyi Zhang and Xinliang Cao; Visualization, Mingliang Che and Jiyi Zhang; Writing—original draft, Fan Yang; Writing—review & editing, Fan Yang, You Li and Chi Zhang. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (no. 42001322), the Open Fund of Key Laboratory of Urban Land Resources Monitoring and Simulation, Ministry of Natural Resources (no. KF-2021-06-022), the Introduction Program of High-Level Innovation and Entrepreneurship Talents in Jiangsu Province (no. JSSCBS20211131) and the Project of Nantong Science and Technology Bureau (no. JC2020174).

Data Availability Statement: Not applicable.

Acknowledgments: The authors acknowledge the Visualization and MultiMedia Lab at University of Zurich (UZH) and the ISPRS WG IV/5 [40] for the acquisition of data. The authors also thank anonymous reviewers for their insightful suggestions and comments.

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

References

1. Bassier, M.; Vergauwen, M. Topology Reconstruction of BIM Wall Objects from Point Cloud Data. *Remote Sens.* **2020**, *12*, 1800. [CrossRef]
2. Li, H. 3D Indoor Scene Reconstruction and Layout Based on Virtual Reality Technology and Few-Shot Learning. *Comput. Intell. Neurosci.* **2022**, *2022*, 4134086. [CrossRef]
3. Nikoohemat, S.; Diakit , A.A.; Lehtola, V.; Zlatanova, S.; Vosselman, G. Consistency grammar for 3D indoor model checking. *Trans. GIS* **2021**, *25*, 189–212. [CrossRef]
4. Kleiner, A.; Baravalle, R.; Kolling, A.; Pilotti, P.; Munich, M. A Solution to Room-By-Room Coverage for Autonomous Cleaning Robots. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017.
5. Kang, Z.; Yang, J.; Yang, Z.; Cheng, S. A Review of Techniques for 3D Reconstruction of Indoor Environments. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 330. [CrossRef]
6. Zlatanova, S.; Isikdag, U. 3D Indoor Models and Their Applications. In *Encyclopedia of GIS*; Shekhar, S., Xiong, H., Zhou, X., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 9–20.
7. Fang, H.; Lafarge, F.; Pan, C.; Huang, H. Floorplan generation from 3D point clouds: A space partitioning approach. *ISPRS J. Photogramm. Remote Sens.* **2021**, *175*, 44–55. [CrossRef]
8. Chen, K.; Lai, Y.-K.; Hu, S.-M. 3D indoor scene modeling from RGB-D data: A survey. *Comput. Vis. Media* **2015**, *1*, 267–278. [CrossRef]
9. Lim, G.; Doh, N. Automatic Reconstruction of Multi-Level Indoor Spaces from Point Cloud and Trajectory. *Sensors* **2021**, *21*, 3493. [CrossRef]
10. Kolluri, R.; Shewchuk, J.R.; O'Brien, J.F. Spectral surface reconstruction from noisy point clouds. In Proceedings of the Geometry Processing (Eurographics/ACM SIGGRAPH), Nice, France, 8 July 2004; pp. 11–21.
11. Kazhdan, M.; Bolitho, M.; Hoppe, H. Poisson Surface Reconstruction. In Proceedings of the Fourth Eurographics Symposium on Geometry Processing, Cagliari, Sardinia, Italy, 26 June 2006; pp. 61–70.
12. Park, S.-J.; Hong, K.-S. Recovering an indoor 3D layout with top-down semantic segmentation from a single image. *Pattern Recognit. Lett.* **2015**, *68*, 70–75. [CrossRef]
13. Chenxi, L.; Schwing, A.G.; Kundu, K.; Urtasun, R.; Fidler, S. Rent3D: Floor-plan priors for monocular layout estimation. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 3413–3421.
14. Zou, C.; Colburn, A.; Shan, Q.; Hoiem, D. LayoutNet: Reconstructing the 3D Room Layout from a Single RGB Image. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2051–2059.
15. Chang, A.; Dai, A.; Funkhouser, T.; Halber, M.; Niebner, M.; Savva, M.; Song, S.; Zeng, A.; Zhang, Y. Matterport3D: Learning from RGB-D Data in Indoor Environments. In Proceedings of the 2017 International Conference on 3D Vision (3DV), Qingdao, China, 10–12 October 2017; pp. 667–676.
16. Mura, C.; Mattausch, O.; Villanueva, A.J.; Gobbetti, E.; Pajarola, R. Automatic room detection and reconstruction in cluttered indoor environments with complex room layouts. *Comput. Graph.* **2014**, *44*, 20–32. [CrossRef]

17. Armeni, I.; Sener, O.; Zamir, A.R.; Jiang, H.; Brilakis, I.; Fischer, M.; Savarese, S. 3D Semantic Parsing of Large-Scale Indoor Spaces. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1534–1543.
18. Luperto, M.; Amigoni, F. Reconstruction and prediction of the layout of indoor environments from two-dimensional metric maps. *Eng. Appl. Artif. Intell.* **2022**, *113*, 104910. [\[CrossRef\]](#)
19. Ochmann, S.; Vock, R.; Klein, R. Automatic reconstruction of fully volumetric 3D building models from oriented point clouds. *ISPRS J. Photogramm. Remote Sens.* **2019**, *151*, 251–262. [\[CrossRef\]](#)
20. Khoshelham, K.; Díaz-Vilariño, L. 3D Modelling of Interior Spaces: Learning the Language of Indoor Architecture. *ISPRS -Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2014**, *XL-5*, 321–326. [\[CrossRef\]](#)
21. Becker, S.; Peter, M.; Fritsch, D. Grammar-Supported 3D Indoor Reconstruction from Point Clouds for “As-Built” BIM. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2015**, *II-3/W4*, 17–24. [\[CrossRef\]](#)
22. Ikehata, S.; Yang, H.; Furukawa, Y. Structured Indoor Modeling. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Washington, DC, USA, 7–13 December 2015; pp. 1323–1331.
23. Murali, S.; Speciale, P.; Oswald, M.R.; Pollefeys, M. Indoor Scan2BIM: Building information models of house interiors. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 6126–6133.
24. Cai, Y.; Fan, L. An Efficient Approach to Automatic Construction of 3D Watertight Geometry of Buildings Using Point Clouds. *Remote Sens.* **2021**, *13*, 1947. [\[CrossRef\]](#)
25. Yang, F.; Zhou, G.; Su, F.; Zuo, X.; Tang, L.; Liang, Y.; Zhu, H.; Li, L. Automatic Indoor Reconstruction from Point Clouds in Multi-room Environments with Curved Walls. *Sensors* **2019**, *19*, 3798. [\[CrossRef\]](#)
26. Mura, C.; Mattausch, O.; Pajarola, R. Piecewise-planar Reconstruction of Multi-room Interiors with Arbitrary Wall Arrangements. *Comput. Graph. Forum* **2016**, *35*, 179–188. [\[CrossRef\]](#)
27. Ambrus, R.; Claiici, S.; Wendt, A. Automatic Room Segmentation From Unstructured 3-D Data of Indoor Environments. *IEEE Robot. Autom. Lett.* **2017**, *2*, 749–756. [\[CrossRef\]](#)
28. Cui, Y.; Li, Q.; Yang, B.; Xiao, W.; Chen, C.; Dong, Z. Automatic 3-D Reconstruction of Indoor Environment With Mobile Laser Scanning Point Clouds. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**, *12*, 3117–3130. [\[CrossRef\]](#)
29. Nikoohemat, S.; Diakité, A.A.; Zlatanova, S.; Vosselman, G. Indoor 3D reconstruction from point clouds for optimal routing in complex buildings to support disaster management. *Autom. Constr.* **2020**, *113*, 103109. [\[CrossRef\]](#)
30. 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. [\[CrossRef\]](#)
31. Ochmann, S.; Vock, R.; Wessel, R.; Klein, R. Automatic reconstruction of parametric building models from indoor point clouds. *Comput. Graph.* **2016**, *54*, 94–103. [\[CrossRef\]](#)
32. Museth, K. VDB: High-resolution sparse volumes with dynamic topology. *ACM Trans. Graph.* **2013**, *32*, 1–22. [\[CrossRef\]](#)
33. Yang, F.; Che, M.; Zuo, X.; Li, L.; Zhang, J.; Zhang, C. Volumetric Representation and Sphere Packing of Indoor Space for Three-Dimensional Room Segmentation. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 739. [\[CrossRef\]](#)
34. Lin, Y.; Li, J.; Wang, C.; Chen, Z.; Wang, Z.; Li, J. Fast regularity-constrained plane fitting. *ISPRS J. Photogramm. Remote Sens.* **2020**, *161*, 208–217. [\[CrossRef\]](#)
35. Oesau, S.; Lafarge, F.; Alliez, P. Planar Shape Detection and Regularization in Tandem. *Comput. Graph. Forum* **2015**, *35*, 203–215. [\[CrossRef\]](#)
36. Botsch, M.; Kobbelt, L.; Pauly, M.; Alliez, P.; Lévy, B. *Polygon Mesh Processing*; CRC Press: Boca Raton, FL, USA, 2010.
37. Boykov, Y.; Veksler, O.; Zabih, R. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23*, 1222–1239. [\[CrossRef\]](#)
38. CGAL, Computational Geometry Algorithms Library. 2022. Available online: <http://www.cgal.org> (accessed on 1 May 2022).
39. Cloud Compare. 3D Point Cloud and Mesh Processing Software Open Source Project. 2022. Available online: <http://www.cloudcompare.org/> (accessed on 1 May 2022).
40. Khoshelham, K.; Tran, H.; Acharya, D.; Vilariño, L.D.; Kang, Z.; Dalyot, S. Results of the ISPRS benchmark on indoor modelling. *ISPRS Open J. Photogramm. Remote Sens.* **2021**, *2*, 100008. [\[CrossRef\]](#)
41. Tran, H.; Khoshelham, K.; Kealy, A. Geometric comparison and quality evaluation of 3D models of indoor environments. *ISPRS J. Photogramm. Remote Sens.* **2019**, *149*, 29–39. [\[CrossRef\]](#)
42. Khoshelham, K.; Vilariño, L.D.; Peter, M.; Kang, Z.; Acharya, D. The ISPRS Benchmark on Indoor Modelling. *ISPRS-Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, *XLII-2/W7*, 367–372. [\[CrossRef\]](#)