ISPRS Journal of Photogrammetry and Remote Sensing 177 (2021) 57–74

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

Vectorized indoor surface reconstruction from 3D point cloud with multistep 2D optimization☆

Jiali Han [a,b,c], Mengqi Rong [a,b,c], Hanqing Jiang [d], Hongmin Liu [e,*], Shuhan Shen [a,b,c,*]

[a] National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[b] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China
[c] CASIA-SenseTime Research Group, China
[d] SenseTime Research, Hangzhou, Zhejiang, China
[e] School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China

ARTICLE INFO

Keywords:
Indoor reconstruction
Vectorized model
Multistep 2D optimization
LoD
Semantic segmentation

ABSTRACT

Vectorized reconstruction from indoor point cloud has attracted increasing attention in recent years due to its high regularity and low memory consumption. Compared with aerial mapping of outdoor urban environments, indoor point cloud generated by LiDAR scanning or image-based 3D reconstruction usually contain more clutter and missing areas, which greatly increase the difficulty of vectorized reconstruction. In this paper, we propose an effective multistep pipeline to reconstruct vectorized models from indoor point cloud without the Manhattan or Atlanta world assumptions. The core idea behind our method is the combination of a sequence of 2D segment or cell assembly problems that are defined as global optimizations while reducing the reconstruction complexity and enhancing the robustness to different scenes. The proposed method includes a semantic segmentation stage and a reconstruction stage. First, we segment the permanent structures of indoor scenes, including ceilings, floors, walls and cylinders, from the input data, and then, we reconstruct these structures in sequence. The floorplan is first generated by detecting wall planes and selecting optimal subsets of projected wall segments with Integer Linear Programming (ILP), followed by constructing a 2D arrangement and recovering the ceiling and floor structures by Markov Random Filed (MRF) labeling on the arrangement. Finally, the wall structures are modeled by lifting each edge of the arrangement to a proper height by means of another global optimization. Merging the respective results yields the final model. The experimental results show that the proposed method could obtain accurate and compact vectorized models on both precise LiDAR data and defect-laden MVS data compared with other state-of-the-art approaches.

In recent years, 3D reconstruction of indoor environments has attracted increasing attention due to its great potential for indoor navigation, Building Information Modeling (BIM), virtual reality, and so on. Compared with outdoor urban reconstruction (Musialski et al., 2013), the structure of indoor environments is more complex and contains a large amount of clutter and occlusions, which make it difficult to reconstruct a complete and accurate model. Over the past few decades, many methods have been developed to reconstruct indoor models, such as 2D architectural drawing-based methods (Lee et al., 2008; Horna et al., 2009; Li et al., 2010). However, it is difficult for these methods to achieve full automation, and the robust reconstruction of indoor scenes with varying complexity is still a challenging task.

LiDAR point cloud and image-based point cloud are two commonly used types of data in 3D reconstruction. The former can be obtained by laser scanning with high precision and high density. However, the point cloud obtained by LiDAR (such as Velodyne) contains only geometry information and the device cost varies greatly with the measurement resolution, measurement accuracy and number of beams. In addition, RGB-D sensors (such as Kinect) are also used to generate dense point cloud, which need low device cost while the measure distance is short and thus it will be cumbersome and time-consuming to scan large-scale scenes. In contrast, Structure-from-Motion (SfM) (Schonberger and Frahm, 2016; Cui et al., 2017) and Multi-View-Stereo (MVS) (Schönberger et al., 2016; Shen, 2013) can generate point cloud from

☆ This work was supported by the National Natural Science Foundation of China (No. 61873265 and 61632003).
* Corresponding author.
E-mail addresses: jiali.han@nlpr.ia.ac.cn (J. Han), mengqi.rong@nlpr.ia.ac.cn (M. Rong), jianghanqing@sensetime.com (H. Jiang), hmliu_82@163.com (H. Liu), shshen@nlpr.ia.ac.cn (S. Shen).

https://doi.org/10.1016/j.isprsjprs.2021.04.019
Received 29 January 2021; Received in revised form 26 April 2021; Accepted 27 April 2021

massive images with rich texture information at a low equipment cost. Besides, the image data can be collected efficiently, such as low-cost crowdsourced data collection, and this data source is what we are concerned more about. Considering that the dense point cloud obtained by different methods always lacks topological relationships and inevitably includes outliers and holes, especially for MVS points, it is more reasonable and more common to represent objects with a triangular mesh surface. Traditional surface reconstruction methods (Kazhdan et al., 2006; Vu et al., 2011) aim to generate surface models with dense facets as accurately as possible. Nevertheless, this representation is still redundant and lacks the structural and semantic information of the scenes, which hinders its applications in many modern systems, such as navigation services that require lightweight model storage. Considering the above problems, lots of approaches have been developed to generate vectorized models that are mainly represented by compact polygons (Previtali et al., 2014; Yu et al., 2021; Tran and Khoshelham, 2019; Thomson and Boehm, 2015; Becker et al., 2015; Wang et al., 2017; Tran et al., 2018; Previtali et al., 2018). In these methods, many (Tran et al., 2018; Previtali et al., 2018) are based on the Manhattan or Atlanta world assumptions, which are common in scene reconstruction. The former term assumes that the scene has only three orthogonal directions, and the latter term defines a vertical direction and a set of directions orthogonal to it, which can represent wider scenes. These assumptions simplify the problem and enhance the model regularity, but they limit the generalization of the methods. In addition, the Levels of Detail (LoDs) defined by CityGML (Kutzner et al., 2020) provides a progressive vectorized reconstruction of models, which has become a data standard for many applications.

This paper proposes a complete multistep pipeline to reconstruct vectorized LoD2 indoor buildings that confirm to CityGML 3.0 (Kutzner et al., 2020), and Fig. 1 is an example of LoDs. Unlike general reconstruction in 3D space, we decompose the 3D reconstruction problem into a sequence of 2D segment or cell assembly problems, and we unify the 3D shape detection and 2D energy optimization into a framework. Considering that indoor scenes always have complex structures and contain a large amount of clutter and many occlusions, we first segment the permanent structures of the scenes, including the floors, ceilings, cylinders and walls, from the input point cloud. Next, we model these structural elements in sequence. Based on the key observations that indoor scenes are mainly composed of piecewise planar structures and the walls are perpendicular to the ground in most scenes, we detect wall planes, project them to the ground plane and generate the floorplan from the wall candidate segments. Then, we construct a 2D arrangement based on the floorplan and perform Markov Random Field (MRF) optimization to reconstruct the floor and ceiling models. The cylinder model is recovered by representing the detected cylinders from input with regular octahedra. Finally, the wall model can be obtained by lifting edges of the 2D arrangement to the correct heights. Merging all of the structural results obtains the final model. The main contributions of our work are as follows:

- Propose a multistep and versatile indoor LoD2 vectorized reconstruction pipeline without the Manhattan or Atlanta world assumptions, taking the 3D shape detection and 2D energy optimization into a uniform framework and having the ability to address various indoor scenes with different characteristics, especially including complex ceiling and floor structures.
- Decompose the 3D reconstruction problem into a sequence of 2D segment or cell assembly sub-problems by means of semantic segmentation, which reduces the complexity of the reconstruction. Each sub-problem is defined as a global optimization that can be solved effectively. A combination of each of the results can generate a reasonable and consistent final model.
- By using multiple energy terms (point supporting, coverage, planarity, etc.) in the optimization function and introducing prior rules as constraints, each stage in the proposed method is designed to have good robustness to noise and missing areas in the point cloud, which makes the proposed pipeline suitable for processing both the precise LiDAR data and defect-laden MVS data.

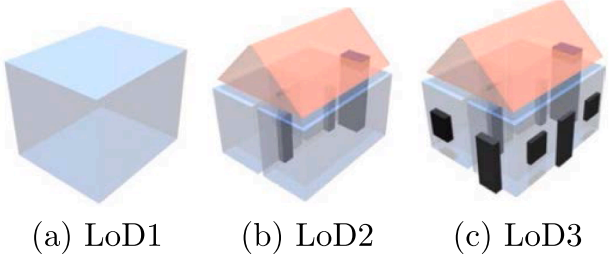


Fig. 1. An example of indoor LoDs (Billen et al., 2012) defined. in CityGML 3.0.

1. Related work

Recently, a considerable amount of work has been performed to reconstruct indoor environments in photogrammetry, computer vision and computer graphics, and detailed reviews can be found in (Pintore et al., 2020; Kang et al., 2020). In addition, an thorough survey of surface reconstruction methods by prior can be found in (Berger et al., 2017). Here, we focus more on the vectorized model surface reconstruction, and we review studies that are relevant to ours, covering polygonal structure modeling, room floorplan estimation and mesh simplification.

1.1. Polygonal structure modeling

Considering that man-made objects such as buildings usually have strong structural characteristics, a large amount of work has focused on extracting the structural elements of indoor scenes and representing models with polygons. Generally, there are two main types of structural elements: geometric elements (*e.g.* plane, line, corner) and semantic elements (*e.g.* wall, roof).*Geometric elements-based modeling.* Since indoor buildings are mainly composed of piecewise planes, many methods extract plane elements from the scene and recover final models by choosing appropriate candidate faces or space cells (Ochmann et al., 2016; Ochmann et al., 2019; Cui et al., 2019; Li et al., 2019; Wang et al., 2020; Tran and Khoshelham, 2020; Oesau et al., 2014; Previtali et al., 2018). Ochmann et al. constructed a planar graph by intersecting wall centerlines, which are obtained by wall candidate plane projections, and performed room labeling on the graph with an energy minimization approach to obtain the final model (Ochmann et al., 2016). This method needed indoor scans to provide initial point cloud segmentation and did not support multi-story buildings. Later, Ochmann et al. solved these problems by applying prior-free Markov Clustering (vanDongen, 2000) to cluster point cloud and intersecting all of the wall and slab candidates in 3D space (Ochmann et al., 2019). With the Manhattan world assumption, Cui et al. detected horizontal ceilings and floors according to the density histogram of the z coordinates and used visibility analysis to segment individual rooms. Finally, the models are reconstructed with multi-label graph cuts (Cui et al., 2019). The above methods converted the modeling problem into a cell labeling problem in which the room segmentation was important information, while they did not consider complex roof and floor structures, such as sloping structures. Discarding the previous assumptions, Nan et al. proposed a general pipeline in which plane elements were detected from the point cloud and used to slice 3D space into convex polyhedra, followed by selecting polygon faces to assemble the final model (Nan and Wonka, 2017). Recently, Bauchet et al. introduced a kinetic data structure to dynamically intersect the 3D space, which led to a lighter yet efficient partition (Bauchet and Lafarge, 2020). Our work is most closely related to this section.

Compared with the above methods, the proposed method in this paper is a general vectorized reconstruction method to deal with point cloud obtained by different devices. we disassembled the 3D reconstruction in indoor environments into segment selection and cell labeling optimization sub-problems in lower 2D space without the Manhattan or Atlanta world assumptions and included the ability to address complex ceiling and floor structures.

Recently, Liu et al. introduced an end-to-end deep network architecture to obtain a wireframe model from point cloud (Liu et al., 2021). They first detected patches that may contain a corner and then predicted corner positions from patches, followed by detecting edges from many pairs of vertices. This method made full use of high-level point features and was the first to generate a vectorized wireframe model through the end-to-end strategy. However, this method was affected by the quality of the point cloud, and the inevitable noise and missing areas in point cloud may reduce the quality of the reconstructed models.*Semantic elements-based modeling.* Generally, indoor environments contain semantic structures such as ceilings, walls and floors in each room that can be used to recover building models. Mura et al. converted point cloud into a semantically richer adjacency graph and selected permanent components based on the structural paths in 3D, followed by volumetric segmentation to reconstruct the rooms (Mura et al., 2016). Ikehata et al. represented the indoor scene as a structure graph, where the nodes corresponded to some elements such as walls, and they conducted reconstructions by applying a sequence of structure grammars (Ikehata et al., 2015).

1.2. Room floorplan estimation

Wall structures are always the focus of indoor reconstruction due to their complexity, and they are usually represented by 2D floorplans, which can be obtained from point cloud (Phalak et al., 2020; Liu et al., 2018; Chen et al., 2019) or images (Zeng et al., 2020; Sun et al., 2019; Yang et al., 2019) and can express vectorized LoD0 models (outline the building footprint). Liu et al. predicted the pixel-wise geometric and semantic information through three network branches, and the floorplan meeting the constraint conditions was obtained through Integer Programming (Liu et al., 2018). They exchanged intermediate features between different branches and made full use of the whole network architectures; however, this method was more suitable for the Manhattan world scenes. Chen et al. discarded the Manhattan limit, inferred the room segmentation and corners/edges likelihood using two networks and obtained the floorplan using a global graph optimization (Chen et al., 2019). Later, Phalak et al. performed room and wall clustering using a deep network and predicted the room perimeters using procedural algorithms on each room individually (Phalak et al., 2020), which was a local problem and could be processed in parallel. Different from using 3D point cloud, Sun et al. took a single-view panoramic image as input, used a recurrent neural network to capture global information and encoded layouts as three 1D vectors, which required fewer computation resources (Sun et al., 2019). Our pipeline can generate a vectorized LoD2 model while also providing the floorplan information. Compared with the above methods, we mainly used the geometric features of the point cloud and were less dependent on the contents and styles of the input data, which, however, is an important problem that may affect the results of the learning-based methods.

1.3. Mesh simplification

Instead of directly generating compact polygon representations, much work has obtained simplified models by reducing the number of facets of dense meshes, especially in computer graphics (Garland and Heckbert, 1997; Cohen-Steiner et al., 2004; Salinas et al., 2015; Bouzas et al., 2020; Li and Nan, 2021). Garland et al. iteratively contracted vertex pairs and tracked surface error approximations using quadric matrices (Garland and Heckbert, 1997). Steiner et al. repeatedly clustered facets into best-fitting regions to reduce the distortion error with the help of geometric proxies (Cohen-Steiner et al., 2004). Salinas et al. performed greedy mesh decimation by edge collapsing while preserving the model structures using planar proxies (Salinas et al., 2015). Compared with direct polygon generation methods, these mesh reduction methods can also yield the compact models; however, the regularity of their results may not be as good as the direct methods.

2. Methods

Fig. 2 illustrates the workflow of the proposed method. The proposed multistep framework takes point cloud obtained by LiDAR or MVS as input and outputs compact polygonal models that confirm to CityGML LoD2 and have no overhang structures. This method consists of two main stages: semantic segmentation and LoD reconstruction. The point cloud of the permanent structures, including ceilings, floors, walls and cylinders, is first segmented from input data by means of normal information or semantic segmentation networks. Then, with the segmented point cloud, the LoD reconstruction can be performed by the following three key steps:

1. *Floorplan generation.* The wall planes are detected from wall point cloud and are projected onto the ground to slice the ground plane into 2D cells. Then, the floorplan can be obtained by selecting a set of cell edges via Integer Linear Programming (ILP).
2. *Non-wall structure reconstruction.* Considering that the ceilings and floors may consist of several planes with different heights and complex structures, we first detect the ceiling and floor planes from the corresponding point cloud and project the plane intersection lines onto the floorplan to construct a 2D arrangement. Then, we perform ceiling and floor plane labeling on the arrangement respectively using MRF optimization. Extruding the arrangement of the cells to their label planes gives the ceiling and floor models. The cylinder model can be obtained by detecting cylinder structures from semantic point cloud and representing them with regular octahedra.
3. *Wall structure reconstruction.* Because the walls may contain different heights, we use the ceiling and floor label assignments and wall point cloud to help restore the wall structure. And the wall model can be obtained by selecting the optimal edges of arrangement cells via ILP and lifting them to proper heights.

Finally, the LoD2 model can be generated by merging the wall and non-wall reconstruction results. In the following sections, these steps will be described in detail.

2.1. Scene segmentation

To obtain vectorized models with semantic information, we segment permanent structures, including ceilings, floors, walls and cylinders, from the input data. First, the indoor point cloud is aligned to the building reference coordinates, where the Z-axis is perpendicular to the ground plane. Then, we use different segmentation methods to address the LiDAR and MVS data.

2.1.1. Segmentation on the LiDAR data

LiDAR data here mainly include pure LiDAR point cloud without image information. Considering the high precision and high density of the LiDAR data, we mainly use the geometric attributes, such as normal information, to segment it. Specifically, the vertical and horizontal point cloud are first separated by selecting points whose normals are nearly perpendicular to the Z-axis and nearly parallel to the Z-axis with the deviation no more than a predefined threshold $thre_{ang}$. Then, we use the K-nearest neighbors (KNN) of each point in the vertical point cloud to compute its curvature and select points whose curvatures are less than $thre_k$ as the wall points with the remaining points taken as cylinder points. After obtaining the wall point cloud, we compute the mean

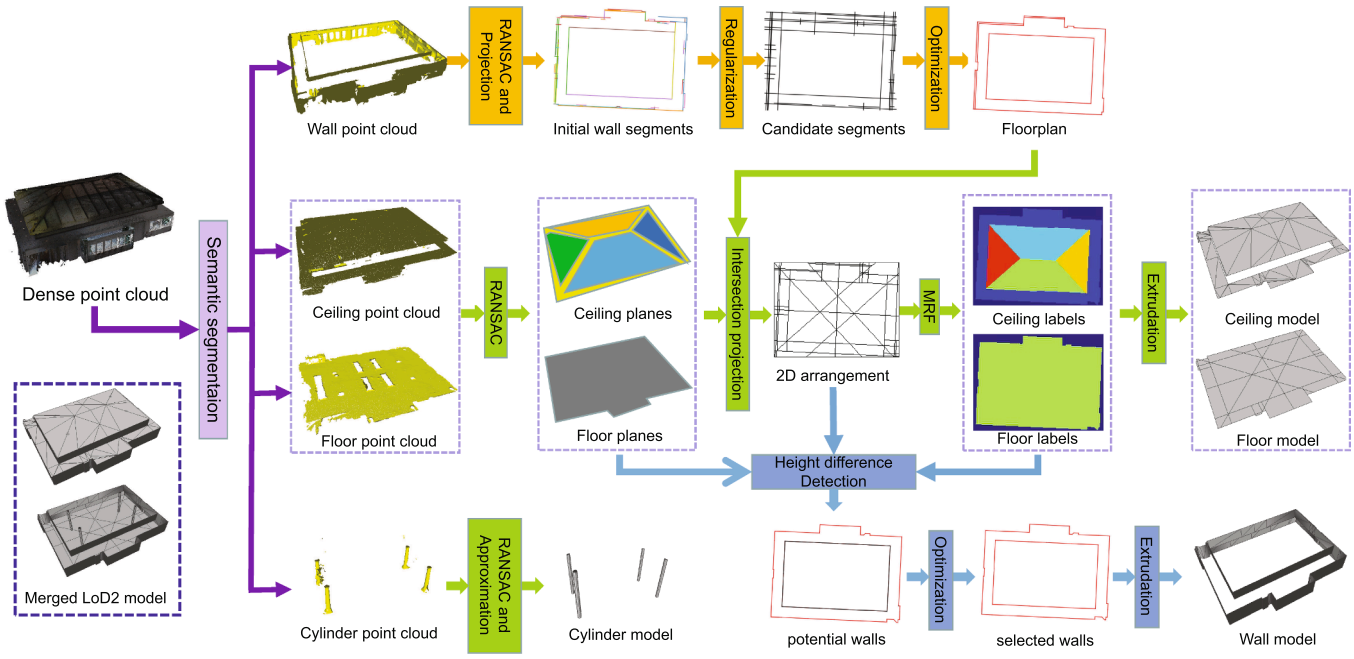


Fig. 2. Overview of the proposed method. There are two stages in the method: semantic segmentation and LoD reconstruction. The latter includes floorplan generation, non-wall structure reconstruction and wall structure reconstruction, which are highlighted with orange, green and blue respectively.

height H of the points and segment the ceiling points whose heights are higher than the H from the horizontal point cloud, with the remaining points taken as floor points. In view of the fact that the ceilings and floors may contain slanted planes, we detect planes from the original point cloud except for the vertical and horizontal points using RANSAC (Schnabel et al., 2007). Then, we compute the average height of the supporting point set that corresponds to each detected plane and add the supporting point sets whose average height is higher than H to the ceiling point cloud; in addition, we add the point sets whose average height is lower than H to the floor point cloud. In our experiments, the threshold $thre_{ang}$ is set to 5°, K is set to 10, and $thre_k$ is set to 1. Fig. 3 displays the segmentation results on a LiDAR dataset used in our experiments. In addition to using the geometric attributes of the LiDAR point cloud to segment the indoor scenes, some 3D convolutional neural networks (CNNs) (Qi et al., 2017a; Qi et al., 2017b) can also be used for point cloud segmentation. However, currently, the 3D CNNs are more suitable for the recognition or segmentation of single small objects, and when addressing large-scale scenes, these 3D networks usually need large amounts of training data for successful generalization, which is a hard labor-intensive task.

2.1.2. Segmentation on the MVS data

Compared with precise LiDAR point cloud, MVS point cloud inevitably contains outliers, noise and missing parts, especially for indoor scenes with a large number of weak texture areas, which makes segmentation using only geometric features unreliable. Considering that the point-meshing algorithm has a certain ability to filter out outliers and repair holes, and that the camera's intrinsic (focal length, principle point, distortions) and extrinsic (camera 6-DOF poses) parameters are calibrated as a by-product during SfM and MVS, we transform the MVS point cloud into a triangular mesh using visibility-based meshing (Vu et al., 2011) in most cases and Poisson Surface Reconstruction (PSR) (Kazhdan et al., 2006) when visibility information is not available, and we perform segmentation on this mesh by making full use of the calibrated images. Here, we adopt the mature 2D image segmentation network, and use a *Segmentation-Fusion* scheme similar to (Zhou et al., 2018) to perform the segmentation on MVS meshes, which considers the 2D semantic results and 3D geometry information. Specifically, we use DeepLabv3 (Chen et al., 2018) pretrained on Cityscapes (Cordts et al., 2016) as our segmentation network and fine-tune it individually on each scene due to the complex structures of indoor scenes. For each scene, the last layer of weights in the network is dropped, and we manually select and annotate dozens of images that cover different areas of the scene and contain different semantic objects with five categories $l^s = \{ceiling, floor, wall, cylinder, others\}$. Then, we use these training data to fine-tune the network, followed by inputting all of the images of the scene into the network to obtain their pixelwise semantic segmentation results. Next, we backproject the 2D segmented results on the mesh using the image calibration parameter and compute the probability that each facet on the mesh belongs to each label according to its received 2D information. Based on the observation that adjacent facets are more likely to have the same label, we transform the 3D segmentation into an MRF optimization. Specifically, we introduce an objective function that includes a data term and a smooth term to evaluate the energy of the 3D mesh segmentation. The data term is defined on each facet as the negative label probability, and the smooth term is defined on each pair of adjacent facets and represents the adjacent similarity measured by the facet normals. The smaller the angle between the normals of adjacent facets, the higher the similarity is. The MRF optimization is to find a label configuration on all of the facets to minimize the above energy function, which corresponds to good segmentation results. This minimization problem can be solved by the graph-cut (Boykov et al., 2001; Boykov and Kolmogorov, 2004) algorithm. After obtaining the facet labels, we segment the mesh into different semantic blocks according to these labels and then uniformly resample each segmented mesh block to obtain different dense point cloud. Fig. 4 displays the segmentation results on an MVS dataset used in our experiments.

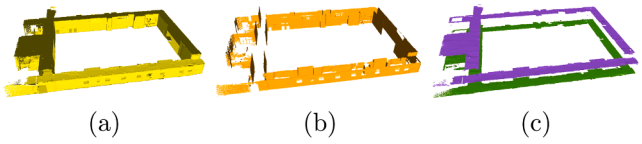


Fig. 3. Scene segmentation on Mimap_bim_00 scene (LiDAR point cloud). (a) is the original point cloud, (b) is the segmented wall point cloud, and (c) is the segmented ceiling and floor point cloud.

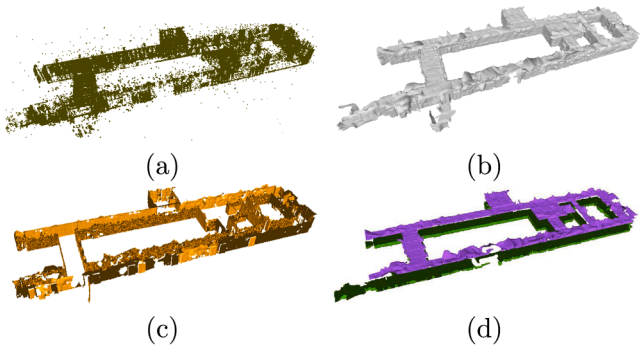


Fig. 4. Scene segmentation on Office_MVS scene (MVS point cloud). (a) is the MVS point cloud, (b) is the reconstructed triangular mesh using visibility-based meshing (Vu et al., 2011, (c) is the segmented wall point cloud, and (d) is the segmented ceiling and fl.oor point cloud.

Note that the quality of the above 3D segmentation is affected by the quality of the 2D image segmentation, which is mainly related to the number of annotated training images. However, depending on redundant information from multiple perspectives and the MRF optimization, the 3D segmentation results are relatively reasonable. In addition, our experiments show that the subsequent LoD reconstruction stage is relatively robust for segmentation, and a certain error in the segmentation result is tolerable, which means that we do not need to manually annotate an excessively large number of images for each scene. In our experiments, it is sufficient to select 25–30 training images on the scene with complex housing patterns, and 5–8 training images are sufficient on the other scenes.

2.2. Floorplan Generation

After obtaining the segmented point cloud, we reconstruct the ceiling, floor, cylinder and wall models in sequence, followed by merging them to generate the final model. Compared with ceilings and floors, the wall structures in the indoor scenes are more difficult to recover due to their complexity. Based on the observation that walls are perpendicular to the ground in most scenes (the same expression with the LoD2 wall in CityGML 3.0 (Kutzner et al., 2020)] and the z coordinates only contain height information, we project the vertical walls onto the ground plane and focus on the floorplan generation in 2D space. Compared with direct extraction in 3D space (Nan and Wonka, 2017), extracting the floorplan in 2D could significantly reduce the computational complexity and improve the robustness. The generated floorplan can greatly assist with the reconstruction of the ceilings, floors and walls. Specifically, there are two core steps in generating the floorplan: wall candidate segment generation and wall segment selection.

2.2.1. Wall candidate segment generation

We detect planes from wall point cloud using RANSAC (Schnabel et al., 2007) implemented in CGAL (The CGAL Project, 2019) with the regularization angle $thre_{ang}$ and leave nearly vertical planes with a deviation of no more than $thre_{ang}$ as wall planes, which are attached with a set of supporting points. Then, we project the detected wall planes onto the ground and project the corresponding 2D supporting points (ignore z coordinates, hereinafter the same.) onto the wall projection lines to generate the wall segments by taking the projected 2D supporting points at the boundary as their vertices, as shown in Fig. 5(a).

After obtaining the wall segments, we extend their vertices and intersect them into more shorter segments with their bounding box as constraints. Due to the noise and missing areas in the point cloud, especially in the MVS data, RANSAC can detect some unsatisfactory planes. Thus, it is necessary to refine the wall segments to obtain a cleaner result with more regularity. Specifically, two wall segments w_i and w_j are to be merged if the following two conditions are met:

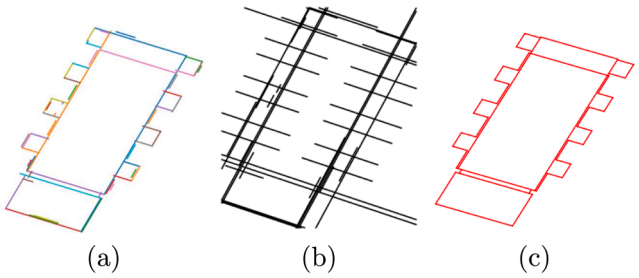


Fig. 5. Floorplan generation on Church_LiDAR scene. (a) is the detected initial wall segments. (b) is the wall candidate segments by extending and regularizing (a). And (c) is the floorplan by selecting the optimal subset of (b) by means of ILP.

1. The angle between the directions of w_i and w_j is less than $thre_{ang2}$
2. The number of common supporting points of w_i and w_j is greater than $thre_{num}(w_i, w_j)$

The merged new segment w_n is determined by passing through a point p_n with the direction $\overrightarrow{w_n}$:

$$p_n = \frac{\| w_i \|}{\| w_i \| + \| w_j \|} Mid\left(w_i\right) + \frac{\| w_j \|}{\| w_i \| + \| w_j \|} Mid\left(w_j\right). \tag{1}$$

$$\overrightarrow{w_n} = \frac{\overrightarrow{w_i} + \overrightarrow{w_j}}{\| \overrightarrow{w_i} + \overrightarrow{w_j} \|}. \tag{2}$$

where $Mid(w_i)$ is the middle point of segment w_i. We merge the sets of the supporting points of segments w_i and w_j as the set of supporting points of segment w_n, and the vertices of segment w_n are computed as described above.

We iteratively regularize the wall segments according to the above criteria until no segment pair satisfies the conditions. Then, the wall candidate segments are obtained by extracting short segments from regularized wall segment intersections, as shown in Fig. 5(b), and we will select a portion of them to generate the floorplan in the next step. In our experiments, the threshold $thre_{ang}$ is set to 5°, the threshold $thre_{ang2}$ is set to 15° and the threshold $thre_{num}(w_i, w_j) = thre_{num} \cdot min(|P(w_i)|, |P(w_j)|)$ where the $thre_{num}$ is set to 10 and the $min(|P(w_i)|, |P(w_j)|)$ is the smaller number of supporting points of segments w_i and w_j.

2.2.2. Wall segment selection

Given a set of wall candidate segments $\{s_i\}$ with their 3D supporting points $\{P(s_i)\}$, we aim to select an appropriate subset of them to generate the closed floorplan via a global energy optimization approach. Specifically, we introduce an integer variable x_i to the ith candidate with the meaning that $x_i = 1$ represents that the candidate is selected and $x_i = 0$ represents that it is not selected, and we define three energy terms similar to (Nan and Wonka, 2017) on each candidate, including the point supporting term S, point coverage term C and model complexity term M. In addition, we define $X = \{x_i\}$ and the total energy function E:

$$E = \lambda_1 \cdot S + \lambda_2 \cdot C + \lambda_3 \cdot M \tag{3}$$

where $\lambda_1, \lambda_2, \lambda_3$ are balance parameters. By solving the segment selection problem by minimizing the above energy formulation, the floorplan is obtained by putting all candidates with $x_i = 1$ together after optimization.*Point supporting in Eq. (3).* This term is used to measure how reliable a wall candidate segment is that is selected, and the higher the support is (which corresponds to the a lower S), the more likely the candidate is to be in the solution:

$$S = 1 - \frac{1}{|P|}\sum_{x_i \in X}\left(\sum_{\substack{\| p, s_i \| < \varepsilon \\ p \in \dot{P}(s_i)}} 1 - \frac{\| p, s_i \|}{\varepsilon}\right)\cdot x_i. \tag{4}$$

where $|P|$ is the total number of supporting points, and $\| p, s_i \|$ is the distance from a point p to a segment s_i. $\dot{P}(s_i)$ is the set of 2D supporting points of s_i. ε is the distance threshold from the point to the segment and is set to the average vertical distance between 2D supporting points to their corresponding wall candidate segments in our experiments.*Point coverage in Eq. (3).* Considering the presence of missing parts in point cloud, this term is introduced to balance the data completeness. Generally, a candidate segment has better quality if its 2D supporting points cover it densely and uniformly. We project the 2D supporting points of segment s_i onto s_i and compute the covered length $len_{cov}(i)$ between the vertices of segment s_i. We define that if the distance between two adjacent projection points is less than the δ, this distance is considered to be covered. Thus, this term is given by:

$$C = \frac{1}{N}\sum_{x_i \in X}\left(1 - \frac{len_{cov}(i)}{len(i)}\right)\cdot x_i \tag{5}$$

where N is the total number of wall candidate segments, and $len(i)$ is the length of segment s_i. In our experiments, the δ is set to five times the density of the supporting points, and the density is set to the average distance between a point and its 10-nearest neighbors.*Model complexity in Eq. (3).* This term is added to generate the floorplan with appropriate complexity. Here, the vertices of the candidate segments are used to measure the complexity. If there are non-collinear segments in all of the selected segments connected by an intersection v_j, as shown in Fig. 6 (b), (d) and (e), we consider that the intersection will introduce a sharp structure and define $Sharp(v_j) = 1$; otherwise, $Sharp(v_j) = 0$. This term is defined as:

$$M = \frac{1}{|V|}\sum_{v_j \in V} Sharp\left(v_j\right) \tag{6}$$

where V is the set of intersections of the wall candidate segments, and $|V|$ is the size of V.*Constraints.* Since we ignore the thickness of the walls, two rooms may be divided by one wall, and a wall may be adjacent to multiple walls. Thus, we discard the 2-manifold constraint and restrict the floorplan to be closed by defining the following:

$$\sum_{s_i \in S_{v_j}} x_i = 0 \; or \; 2 or \; 3 \; or \; 4, \; \forall v_j \in V \tag{7}$$

where S_{v_j} is the set of segments connected to the intersection v_j. The different selections of segments connected by an intersection are displayed in Fig. 6.*Optimization.* We minimize Eq. (3) with Eq. (7) as a constraint by means of the SCIP solver (Gamrath et al., 2020). Then, the floorplan is obtained by selecting wall candidate segments with $x_i = 1$, as shown in Fig. 5(c). The pseudo code for the optimization is shown in **Algorithm 1**.

Algorithm 1. Minimize the energy function E (Eq. (3))

Input: wall candidate segments $\{s_i\}_{i=1...n_s}$, their supporting points set $\{P(s_i)\}$, and the segment intersections $\{v_i\}_{i=1...n_p}$. Output: the selected segment set S.

1: Define n_s data terms of point supporting as in Eq. (4);
2: Define n_s data terms of point coverage as in Eq. (5);
3: Define n_p data terms of model complexity as in Eq. (6);
4: Define n_p constraint terms of model closure as in Eq. (7);
5: Construct the minimization function E as in Eq. (3) with $2n_s + n_p$ data terms and n_p constraint terms, which could be transformed into a constrained 0–1 liner programming problem;
6: Solve this 0–1 linear programming problem using the SCIP solver;
7: Insert all segments $\{s_i\}$ that are labeled as 1 to the set S.

2.3. Non-wall structure reconstruction

The floorplan reflects the outline of the facades in the indoor scenes and can provide rough boundaries of ceilings and floors. Thus, in this section we focus on the reconstruction of non-wall structures including ceilings, floors and cylinders, using the generated floorplan.

In real indoor scenes, the ceilings and floors may contain some non-horizontal planes, such as slant roofs and steps on the floor. The intersections of these planes (*e.g.* roof ridge lines, floor step lines) are critical elements for describing the building contours. However, these elements are usually missing in the floorplan which mainly contains the facade lineaments. Considering this problem, we detect planes using RANSAC (Schnabel et al., 2007) from the ceiling and floor point cloud, and we compute the 3D intersection lines of the ceiling planes $\{P_c\}$ and floor planes $\{P_f\}$, respectively, followed by projecting those lines onto the floorplan. Then, the 2D arrangement is obtained by properly extending the line segments of the floorplan with its bounding box as the boundary, as shown in Fig. 7(a). The segment extending length is set to one tenth of the diagonal length of the bounding box in our experiments. At this point, the obtained 2D arrangement contains more complete contours of ceilings and floors, which assists with the reconstruction of the ceilings and floors. In this section, we will use multilabel MRF optimization to obtain the ceiling plane and floor plane label assignments on this 2D arrangement, respectively. Additionally, the final ceiling and floor models are generated by extruding each cell of the arrangement to its label planes.

2.3.1. Ceilings and floors reconstruction

Algorithm 2. Minimize the energy function E_c (Eq. (8))

Input: 2D arrangement cells $\{c_i\}_{i=1...n_a}$ with n_d pairs of adjacent cells, planes $P = \{P_c\}_{c=1...n_c} \cup blank$ and the ceiling points set $\{H(c_i)\}$.
Output: the plane label set L for cells $\{c_i\}$.

1: Define $n_a * (n_c + 1)$ data terms as in Eq. (9);
2: Define n_d smooth terms as in Eq. (10);
3: Construct the minimization function E_c as in Eq. (8) with $n_a * (n_c + 1)$ data terms and n_d smooth terms, which could be transformed into a multi-label energy optimization problem;
4: Solve this multi-label energy optimization problem using $\alpha - \beta$ swap algorithm;
5: Insert assigned labels for all cells $\{c_i\}$ into the set L.

Ceiling arrangement labeling We ortho-sample the ceiling point cloud with the step size s, which is set to 0.02 m in our experiments, and we

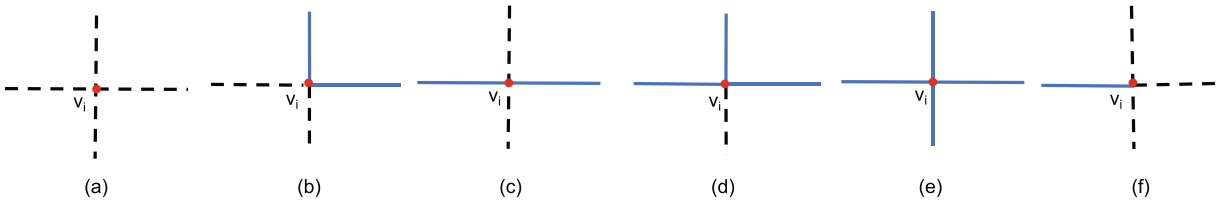


Fig. 6. Different conditions of segment selection. There are totally six conditions when selecting segments connected with an intersection v_i. Due to ignoring the thickness of walls, a wall segment may divide multirooms (corresponds to (d) and (e)), and thus the floorplan is not 2-manifold. However, the floorplan should be guaranteed to be closed and (f) is impossible. Consequently, in the above six conditions, except (f), the others are allowed in our pipeline.

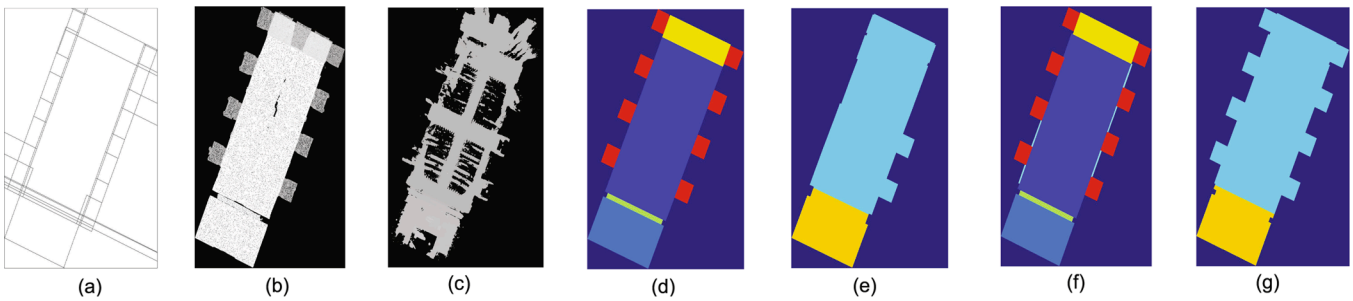


Fig. 7. Ceiling and floor label assignments on Church_LiDAR scene. (a) is the 2D arrangement by adding the projection of intersection lines between different ceilings and different floors on the floorplan. (b) and (c) are ceiling and floor height maps respectively. (d) and (e) are initial labeling results of ceilings and floors. And (f) and (g) are final label assignments after performing the *Label consistency adjustment* in Section 3.3.1. As can be seen, the floor height map (c) has serious deficiency and the initial labeling result (e) has a poor quality. However, with the help of *Label consistency adjustment*, we can obtain a more consistent and complete label result (g), enhancing the robustness of the proposed method to missing areas in the point cloud to a certain extent.

compute the average heights of the points in each grid to form the ceiling height map, as shown in Fig. 7(b). Then, we construct a graph $\mathscr{G} = \{\mathscr{V}, \mathscr{E}\}$, where $\mathscr{V} = \{v_i\}$ is the set of nodes that relate to the cells of the 2D arrangement, and $\mathscr{E} = \{e_i\}$ is the set of edges that relate to the connections of the adjacent cells, and the problem aims to assign each cell c_i to one of the ceiling planes $\{P_c\}$ IDs or the *blank* ID (which represents the outer space). We recover a 3D point p_k for each grid in the height map, and for each cell c_i, we collect points whose projections lie in this cell to form a set $H(c_i)$. The energy function $E_c(L_c)$ with the assignment $L_c = \{l_{v_i}\}$ is defined as:

$$E_c\left(L_c\right) = \sum_{v_i \in \mathscr{V}} D_{v_i}\left(l_{v_i}\right) + \beta \cdot \sum_{e_{v_i,v_j} \in \mathscr{E}} S_{e_{v_i,v_j}}\left(l_{v_i}, l_{v_j}\right) \tag{8}$$

where the former is the data term, the latter is the smooth term and β is the balance parameter, which is set to 0.5 in our experiments.

In Eq. (8), the data term is introduced to measure the fitting degree of set $H(c_i)$ in cell c_i to a plane, which is given by:

$$D_{v_i}\left(l_{v_i}\right) = \frac{|c_i|}{|H(c_i)|} \sum_{p_k \in H(c_i)} ||p_k, l_{v_i}|| \tag{9}$$

where $|c_i|$ is the area of cell c_i, $|H(c_i)|$ is the size of $H(c_i)$, and $||p_k, l_{v_i}||$ is the vertical distance between point p_k to the plane with the label l_{v_i}.

In addition, for penalizing the different label assignments of adjacent cells c_i and c_j, the smooth term is used and is defined as:

$$S_{e_{v_i,v_j}}\left(l_{v_i}, l_{v_j}\right) \begin{cases} |e_{v_i,v_j}| \cdot \left(1 - \dfrac{3\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2}\right) & \text{if} l_{v_i} \neq l_{v_j} \\ 0, & \text{if} l_{v_i} = l_{v_j} \end{cases}, \tag{10}$$

where $|e_{v_i,v_j}|$ is the length of edge e_{v_i,v_j}, which is shared by cells c_i and c_j, λ_0 is the minimum eigenvalue of the covariance matrix of point set $\{p_k | p_k \in H(c_i) \cup H(c_j)\}$, and the equation $\frac{3\lambda_0}{\lambda_0+\lambda_1+\lambda_2}$ can be used to measure the planarity of a point set.

The global optimization problem defined in Eq. (8) is solved by the graph-cut algorithm (Boykov et al., 2001; Boykov and Kolmogorov, 2004), and the generated ceiling label assignment is shown in Fig. 7(d). The pseudo code for the optimization is shown in **Algorithm 2**.*Floor arrangement labeling.* The floor label assignment is obtained by constructing and solving a global optimization function $E_f(L_f)$, which is the same as Eq. (8), in which the ceiling point cloud is replaced with the floor point cloud and the assignment L_c is changed with $L_f = \{l_{v_i}\}$ and $l_{v_i} \in \{P_f\} \cup blank\,ID$. The floor label assignment is shown in Fig. 7(e). *Label consistency adjustment.* Because most of the indoor scenes have closed structures, each area in the scene should have both a ceiling and a floor. However, there may be some missing areas in segmented point cloud, such as Fig. 7(c), and the ceiling and floor label assignments are obtained separately; therefore, for some cells in the 2D assignment, there may be only the ceiling or only the floor. These inconsistent results lead to opening structures, which not only conflict with the closure of the indoor scene but also increase the uncertainty of the subsequent wall reconstruction. To address this problem, we use a label consistency adjustment step to obtain the consistent label assignments.

There are a total of four kinds of results for the ceiling and floor labeling of a cell, as shown in Fig. 8. Among these four results, Fig. 8(b) represents that the cell C_i belongs to the indoor scene and has a ceiling and a floor, and Fig. 8(c) represents that the cell C_i belongs to the outer space, not the indoor scene. The two conditions do not introduce openings and thus are acceptable. In contrast, the last two conditions, Fig. 8d) and and Fig. 8(e), either only have a floor or only have a ceiling, which cause an open side of the scene and thus are not allowed. Here, we aim to adjust the cell labels to solve the inconsistencies. Specifically, we define the total energy $E(L)$ of the cells in the arrangement as:

$$E(L) = E_c(L_c) + E_f(L_f) \tag{11}$$

where $E_c(L_c)$ and $E_f(L_f)$ are the ceiling and floor labeling energy defined in Eq. (8), and $L = L_c \cup L_f$. For each cell with inconsistent labels (corresponding to the conditions (d) and (e) in Fig. 8), we perform the label adjustment in sequence. Taking (d) in Fig. 8 in that a cell has a floor but no ceiling as an example, we change the *blank* ceiling ID to each of $\{P_c\}$ and then compute the energy $E(L)$. Next, we change the floor label P_{fi} to the *blank* floor ID and also compute the energy $E(L)$, followed by selecting the label assignment with the minimum $E(L)$ as the new label of the cell C_i. We apply the above adjustment on all of the cells with inconsistent labels one by one and finally obtain the consistent label assignments. An example is shown in Fig. 7, in which some inconsistent cells in (d) and (e) are successfully modified in (f) and (g). Extruding cells to their label planes gives the ceiling and floor models, as shown in

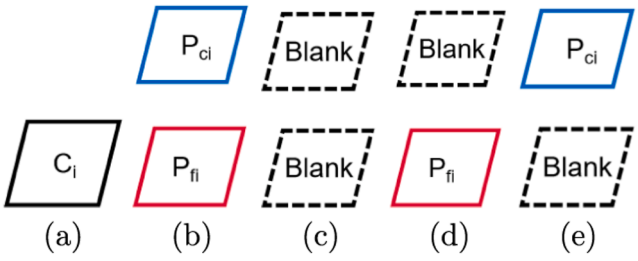


Fig. 8. Example of label assignments on a cell C_i (a). In (b), (c), (d) and (e), the top one is the ceiling plane labeling result of C_i, and the bottom one is the floor plane labeling result of C_i, In order to obtain closed indoor models, (b) and (c) are acceptable, and (d) and (e) are not allowed.

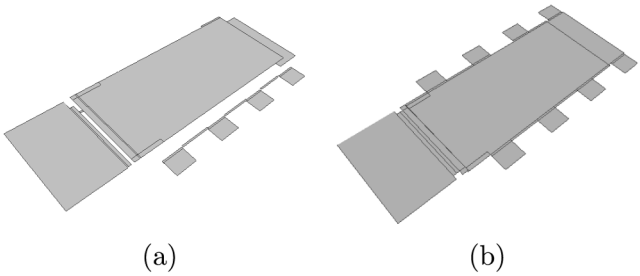


Fig. 9. Ceiling and floor models of Church_LiDAR scene. (a) and (b) are ceiling and floor models by extruding cells of 2D arrangement ((a) in Fig. 7)) to their corresponding planes according to the assignments (f) and (g) in Fig.. 7.

Fig. 9. Note that the label adjustments are performed in sequence, which is a greedy process. However, because the number of cells with inconsistent labels is small and these cells are usually far apart, we found that this simple greedy adjustment is sufficient to obtain consistent and accurate label assignments in our experiments.

By using the above label consistency adjustment, the proposed method has good tolerance to the quality of the point cloud and the semantic segmentation. In addition, different from the assumption that indoor scenes have horizontal ceilings and floors (Cui et al., 2019; Wang et al., 2020; Tran and Khoshelham, 2020), we construct the 2D arrangement with rich contours of scenes and obtain the ceiling and floor models by energy optimization on the arrangement. This strategy can identify and recover the non-horizontal structures (*e.g.* slant roofs, step on the floor) in the ceilings and floors, making the proposed method more flexible in addressing indoor scenes with different structures and complexity.

2.3.2. *Cylinders reconstruction*

In addition to the wall, ceiling and floor, the cylinder is also a common permanent structure in indoor scenes. However, not all scenes contain cylinder structure, and we will perform cylinders reconstruction if a cylinder point cloud is found in the scene segmentation step in Section 3.1.

Since we aim to reconstruct vectorized models with piecewise planes, we detect cylinder structures using RANSAC (Schnabel et al., 2007) from cylinder point cloud and then use regular octahedra to approximate the detected cylinder structures. The heights of the cylinders are determined by the recovered ceiling and floor models.

2.4. *Wall structure reconstruction*

Although we have obtained the ceiling and floor models, not all walls are connected from a floor to a ceiling, as shown in Fig. 10, and thus, it is not feasible to recover the wall model by simply lifting the obtained floorplan (*e.g.* Fig. 5(c)) to the heights of the ceilings and floors. Considering that the 2D arrangement (*e.g.* Fig. 7(a), which contains the floor step line.) contains more complete contours than the floorplan, we view each edge of the arrangement as a potential wall and focus on recovering the wall structures using the arrangement.

According to the ceiling and floor labels of the adjacent cells connected by an edge, there are five valid height conditions, as shown in Fig. 11. Under conditions in which the ceiling or floor label planes of adjacent cells are not connected, such as Fig. 11(c)(d)(e), and in which the edge is on the boundary, such as Fig. 11(f), the height difference is introduced (marked with black segments in Fig. 11). The edges with these height differences are added to the initial wall solution $Wall_i$ for the closure of the models, as shown in Fig. 12(a)(c), and whether to select the edges with other height differences, such as red dashed lines in Fig. 11, is uncertain due to the noise and missing areas in the point cloud, as well as the observation that not all walls are connected from a floor to a ceiling. For these edges $\{e_i\}$ with the height difference

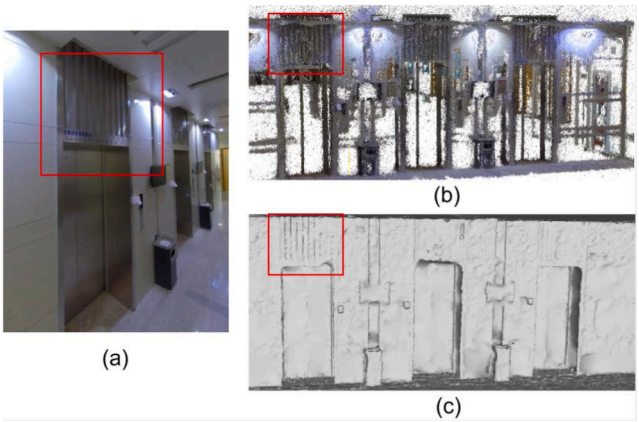


Fig. 10. One close-up on Office_MVS scene. (a) is an image of elevators on Office dataset. (b) is corresponding dense MVS point cloud with color. (c) is corresponding MVS mesh. Due to the existence of elevators, the walls are not connected from a floor to a ceiling directly, such as the area circled in a red rectangle.

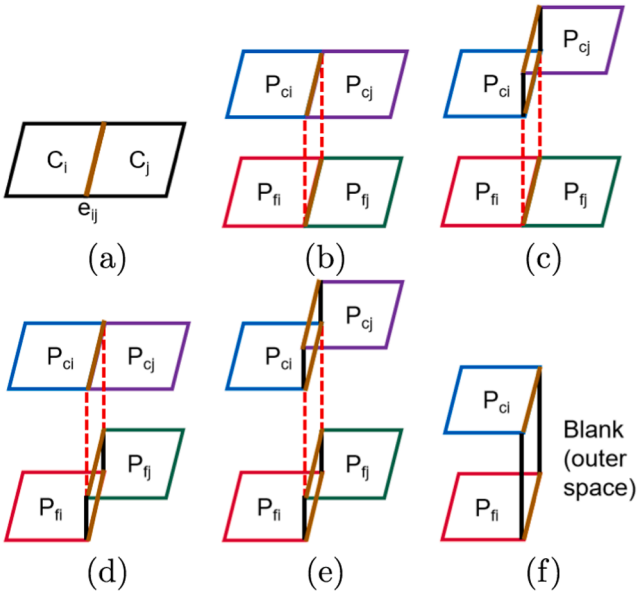


Fig. 11. Different height difference conditions. (a) is a pair of adjacent cells c_i and c_j connected by an edge e_{ij}. In all subpart, the edge e_{ij} is marked with a brown segment. P_{ci} and P_{cj} are ceiling planes and P_{fi} and P_{fj} are floor planes. In (c)-(e), the ceiling label planes or floor label planes of adjacent cells are not connected, introducing the height differences (black segments). In (f), the cell c_j is labeled blank and the edge e_{ij} becomes a boundary, also causing the height difference. For the closure of the final model, the edge e_{ij} with the above height differences should be recovered, and for edge e_{ij} with other height differences (red dashed segments), whether to recover them is an uncertain problem which we solve with an global energy optimization.

$\{(dmax_i, dmin_i)\}$, as shown in Fig. 12(b)(c), we named them $E_{uncertain}$ and introduced the variable $X = \{x_i\}$ for each, which has the same meaning as that used in Eq. (3). We defined the point supporting term S and point coverage term C and cast the selection problem as another energy optimization similar to Eq. (3). The energy function is given by:

$$E = \beta_1 \cdot S + \beta_2 \cdot C \tag{12}$$

where β_1 and β_2 are the balance parameters. Note that if the edge e_i is part of the floorplan, it is attached with a set of supporting points; we retain the points in this set whose z coordinates are within $(dmax_i, dmin_i)$

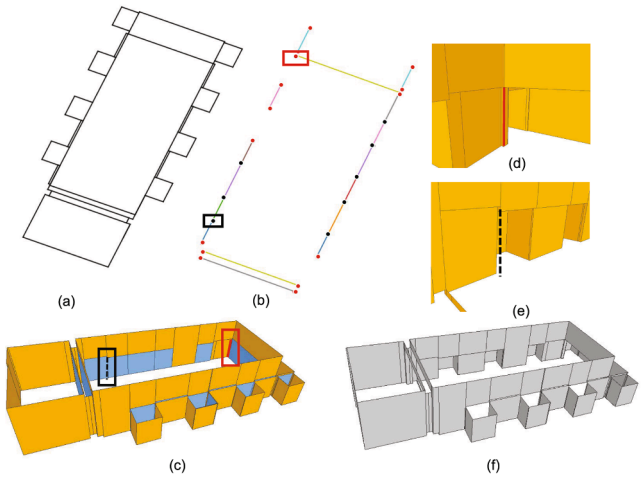


Fig. 12. Wall structure reconstruction on Church_LiDAR scene. (a) is the 2D projection of the initial wall solution $Wall_i$. (b) is the uncertain edges $E_{uncertain}$ with the boundary vertices marked by red dots and others marked by black dots. In (c), the orange walls are initial wall solution $Wall_i$ and the blue walls are uncertain walls by lifting edges in (b) to their corresponding height difference. (d) and (e) are two close-ups of $Wall_i$. The red existed wall edge in (d) corresponds to the boundary vertex circled by red rectangle in (b). And the black uncertain wall edge in (e) corresponds to the non-boundary vertex circled by black rectangle in (b). (f) is the final wall model after optimization.

as the new set $P(e_i)$ of the edge e_i, and otherwise, the set of support points is set to an empty set.*Point supporting in Eq. (12).* This term encourages the selection of edges with more supporting points and is defined as:

$$S = 1 - \frac{1}{|P|}\sum_{x_i \in X} |P\left(e_i\right)| \cdot x_i. \tag{13}$$

where $|P|$ is the total number of supporting points of all edges, and $|P(e_i)|$ is the number of supporting points of edge e_i.*Point coverage in Eq. (12).* This term is used to measure how well the supporting points cover the face f_i, which is constructed by lifting the edge e_i to the height difference $(dmax_i, dmin_i)$. The term is thus defined as:

$$C = \frac{1}{|X|}\sum_{x_i \in X}\left(1 - \frac{\widehat{A}}{A_i}\right) \cdot x_i \tag{14}$$

where $|X|$ is the total number of variables, $\widehat{A}$ is the α-shape area of the projection points from the supporting points of edge e_i to face f_i, and A_i is the area of face f_i. In our experiments, $\sqrt{\alpha}$ is set to the same value as the δ used in Eq. (5).*Constraints.* To ensure the closure of indoor models, the generated walls should be continuous. For the vertex of edge e_i with the height difference $(dmax_i, dmin_i)$, we define it to be a boundary vertex if the height difference $(dmax_i, dmin_i)$ at this vertex already exists in the initial solution $Wall_i$, as shown in Fig. 12(b)(d), where the red dots circled by a red rectangle is a boundary vertex because its corresponding wall edges in (d) (red segment) has existed in $Wall_i$. In addition, the number of selected edges connected by this vertex in this step is unlimited. For non-boundary vertices (black dots in Fig. 12(b)), the number of selected edges connected by them in this step is restricted to not be one to avoid generating discontinuous walls. Thus, the constraint is:

$$\sum_{e_i \in E_{v_j}} x_i = 0 \; or \; 2 or \; 3 \; or \; 4, \; \forall v_j \notin \mathscr{V}_{border} \tag{15}$$

where E_{v_j} is the set of edges connected by vertex v_j, and $\mathscr{V}_{border}$ is the set of boundary vertices.

Optimization. We solve the energy minimization problem defined by Eq. (12) with the constraint Eq. (15) using the SCIP solver (Gamrath et al., 2020), and select the edges with x_i=1 into the initial model $Wall_i$ to generate the complete wall model $Wall_f$, as shown in Fig. 12(f). The pseudo code for the optimization is shown in **Algorithm 3**.

Algorithm 3. Minimize the energy function E (Eq. (12))

Input: edges $\{e_i\}_{i=1...n_e}$ with uncertain heights and their supporting points set $\{P(e_i)\}$.
Output: the selected edge set W.
1: Define n_e data terms of point supporting as in Eq. (13);
2: Define n_e data terms of point coverage as in Eq. (14);
3: Define n_p constraint terms of model closure as in Eq. (15);
4: Construct the minimization function E as in Eq. (12) with $2n_e$ data terms and n_p constraint terms, which could be transformed into a constrained 0–1 liner programming problem;
5: Solve this 0–1 linear programming problem using the SCIP solver;
6: Insert all edges $\{e_i\}$ that are labeled as 1 to the set W.

2.5. LoD2 model assembly

After obtaining the individual models of the permanent structures in indoor scenes, the vectorized LoD2 model with semantic information is generated by simply merging all of the individual results. Due to the ignorance about the wall thickness, the final model is not guaranteed to be a 2-manifold, but it is closed and intersection-free.

In the whole pipeline, we detect planes in 3D space, and most of the remaining work is performed in 2D space without the Manhattan or Atlanta world assumptions, which reduces the problem complexity and enhances the scalability of the proposed method. In addition, the *Label consistency adjustment* following Eq. (11) and the *optimization* following Eq. (12) are helpful in resisting the noise and missing areas in the semantic point cloud, which make our method more robust to the quality of the data and the segmentation.

3. Experimental results

We evaluate the proposed vectorized indoor surface reconstruction method in terms of correctness, simplicity and efficiency, and we compare it with other state-of-the-art methods on several datasets quantitatively or qualitatively. The correctness is measured by the mean Hausdorff distance from the reconstructed model to the ground truth, the simplicity is measured by the number of facets of the final model, and the efficiency is measured by the running time. Our system is implemented in C++ with the CGAL library (The CGAL Project, 2019), the maxflow library (Boykov et al., 2001; Boykov and Kolmogorov, 2004) and the SCIP solver (Gamrath et al., 2020). All of the experiments were performed on a PC with a 4-core Intel Xeon CPU (3.7 GHz) and an NVIDIA Titan RTX GPU.

3.1. Datasets

We use three indoor scene datasets with different characteristics and complexity to comprehensively evaluate our approach. The first dataset is derived from two indoor scenes named *Meeting room* and *Church*, provided by the Tanks and Temples benchmark (Knapitsch et al., 2017), which is used to evaluate the image-based 3D reconstruction methods. We name this dataset the MC dataset, which contains two LiDAR point cloud ground-truths that were captured using an industrial laser scanner (Meeting_room_LiDAR and Church_LiDAR) and two MVS point cloud reconstructed by COLMAP (Schonberger and Frahm, 2016) (Meeting_room_MVS and Church_MVS), as well as the corresponding scene image sets. The second is the BIM feature extraction dataset provided by the MiMAP benchmark (Wen et al., 2020; Wang et al., 2018). This dataset includes three indoor scene LiDAR point cloud, and it describes a closed-loop corridor (Mimap_bim_00), a corridor and multiple rooms (Mimap_bim_01), and a closed-loop corridor and multiple rooms (Mimap_bim_02). The last is our own dataset (Office), which includes a complete office floor scene (Office_MVS). The Office dataset contains an

MVS point cloud generated by COLMAP (Schonberger and Frahm, 2016) from 8586 images with a resolution 800*1,600. The proposed method is performed and compared with other state-of-the-art methods (Nan and Wonka, 2017; Garland and Heckbert, 1997; Salinas et al., 2015; Cohen-Steiner et al., 2004; Liu et al., 2018; Chen et al., 2019) on the above three datasets. The correctness, simplicity and efficiency are evaluated on the first two datasets. Due to the lack of ground truth on the last dataset, we mainly qualitatively validate the robustness of the proposed method on it, in which obvious noise and missing areas exist. Detailed information about the datasets is shown in Table 1.

3.2. Semantic segmentation

In our pipeline, we use a two-stage strategy that includes segmentation and reconstruction to obtain the vectorized indoor models. In the segmentation stage, we segment the permanent structures, including walls, floors, ceilings and cylinders, from the input data and adopt different segmentation methods according to the data types and data characteristics. Specifically, for the MiMAP dataset that only includes LiDAR point cloud with horizontal ceilings and floors and without cylinders, we follow the strategy in Section 3.1.1. First, the normal information is used to segment the vertical and horizontal point cloud using the $thre_{ang} = 5^{\circ}$. Then, we take the vertical point cloud as the wall point cloud because there are no cylinders in the scenes, and we compute the average height of the wall points, followed by using the height to divide the horizontal point cloud into the ceiling and floor point cloud. The segmentation results are shown in Fig. 3 and Fig. 13(e)(f).

For the MC dataset, due to the complex structures of the Meeting_room_LiDAR and Church_LiDAR (include the complex roof beams and lots of clutter), the segmentation results using geometric attributions of point cloud are poor. Thus, we first align the MVS point cloud (Meeting_room_MVS and Church_MVS) to the LiDAR point cloud (Meeting_room_LiDAR and Church_LiDAR) using the transformation matrix provided by the benchmark (Knapitsch et al., 2017), followed by performing Poisson surface reconstruction on LiDAR point cloud and visibility-based meshing on MVS point cloud to obtain the dense mesh models. Due to the strict alignment of the MVS and LiDAR data, the MVS information including the camera parameters, can also be used on the mesh derived from the LiDAR point cloud. Therefore, we adopt the *Segmentation-Fusion* scheme, as mentioned in Section 2.1.2, to perform segmentation on both the MVS and LiDAR meshes. Because the scenes on this dataset have regular rectangular foot areas and contain rich texture information, most of the images in the corresponding scene image set cover a large area of the scene with adequate features. Thus, in our experiments, 5–8 training images on each of the two scenes are sufficient to fine-tune the pretrained network. Fig. 14(a) and (b) show some 2D segmentation results on MC dataset. After obtaining the final semantic mesh, we densely resample the facets on the mesh to obtain the point cloud with different semantic information, as shown in Fig. 13(a)(b)(c)(d). As seen in Fig. 14, the 2D segmentation results are generally acceptable, although there still be a few errors in some local details. By performing the MRF optimization on the 3D mesh, the final segmentation quality is improved, as seen in Fig. 13. However, due to noises and missing areas in the point cloud, there are still some defects in the 3D segmentation results. However, since our subsequent vectorized modeling process is relatively robust to data defects, a certain error in the segmentation results is tolerable.

Table 1
Overview of the different datasets. *#points* is the number of points in the point cloud; *Areas* is the floor area of the bounding box of the scene; t_s is the segmentation time; t_r is the reconstruction time, which consists of the floorplan generation time t_{r1}, non-wall structure reconstruction time t_{r2} and wall structure reconstruction time t_{r3}.

Dataset	Scene	*#points*	*Areas* (m^2)	t_s (sec)	t_r ($t_{r1}/t_{r2}/t_{r3}$) (sec)
MC	Meeting_room_LiDAR	3 M	237.6	220.0	199 (53/134/12)
	Church_LiDAR	3 M	1851.2	166.4	200 (58/130/12)
	Meeting_room_MVS	551 k	237.6	195.0	100 (64/24/12)
	Church_MVS	16 M	1851.2	180.0	185 (40/130/15)
MiMAP	Mimap_bim_00	2.09 M	1528.0	2.3	90 (48/38/4)
	Mimap_bim_01	8.68 M	518.6	9.2	733 (651/33/49)
	Mimap_bim_02	18.65 M	2306.6	19.7	885 (559/171/155)
Office	Office_MVS	100 M	965.7	600.0	259 (201/30/28)

For the Office dataset, we adopt the same strategy with the MC dataset to obtain the semantic point cloud. Since this dataset contains a closed-loop corridor scene, there are smaller valid covering spaces in each image than on the MC dataset. Thus, we pick up 25–30 images on this dataset to fine-tune the pretrained network. Fig. 14(c) shows some 2D segmentation results on Office dataset. In addition, it should be noted that the Office_MVS scene in this dataset contains a large number of weak texture areas, which lead to the serious missing parts in MVS point cloud. The triangular mesh also has some obvious deficiencies even after mending some small holes, as shown in Fig. 23. The existence of massive defects in the Office_MVS makes the reconstruction task more challenging, but it is in line with the real indoor reconstruction situation.

In addition, we record the segmentation time (including the 2D image segmentation time and 3D MRF fusion time if using the *Segmentation-Fusion* scheme.) in Table 1. With the combination of a mature 2D segmentation network and 3D MRF optimization, the *Segmentation-Fusion* scheme can obtain good segmentation results by annotating only dozens of training data.

3.3. Modeling parameter study

The proposed multistep framework contains several key parameters, including plane detection parameters, regularization parameters, wall segment selection weights and wall reconstruction weights. In this section, we perform experiments to show that most of these parameters are insensitive to different scenes, and only a few of them need to be adjusted according to the scenes. Table 2 lists the above-mentioned parameters, and a detailed discussion about them is as follows.*Plane detection parameters.* We detect plane primitives from the wall, ceiling and floor point cloud using RANSAC algorithm due to its robustness to outliers and noise, as is mentioned in (Xia et al., 2020; Kaiser et al., 2019). RANSAC includes the four most important parameters. The first important parameter is p_{min}, which represents the least point number to support a plane, and the smaller its value is, the more planes are detected, and the more noise that could also be generated. In our experiments, we set p_{min} to 500 on Mimap_bim_00 and 1000 on other datasets when detecting wall planes, and we set it to 10000 on all of the datasets to detect the ceilings and floors, which usually consist of more dense and larger planes. The second parameter is d_{max}, which represents the maximum distance between an inlier and a plane, which is set to 0.05 m in our experiments. The third parameter is P_{miss}, which reflects the probability of missing the largest planes; it is set to 0.001 in our experiments. The last parameter is $thre_{dev}$, which controls the maximum deviation of points in one plane, and we set it to 0.90 on the LiDAR data and to 0.85 on the MVS data. Among the above four parameters, p_{min} and $thre_{dev}$ are slightly adjusted to better fit the point cloud with different quality and to obtain more reliable detection results, which is conducive to the subsequent reconstruction. Compared with them, the remaining two parameters have less impact on the detection results, and we set them to the same value on different indoor scenes.*Regularization parameters.* We use the threshold $thre_{ang}$ in Section 3.2.1 when performing RANSAC to regularize the parallelism and orthogonality of the detected

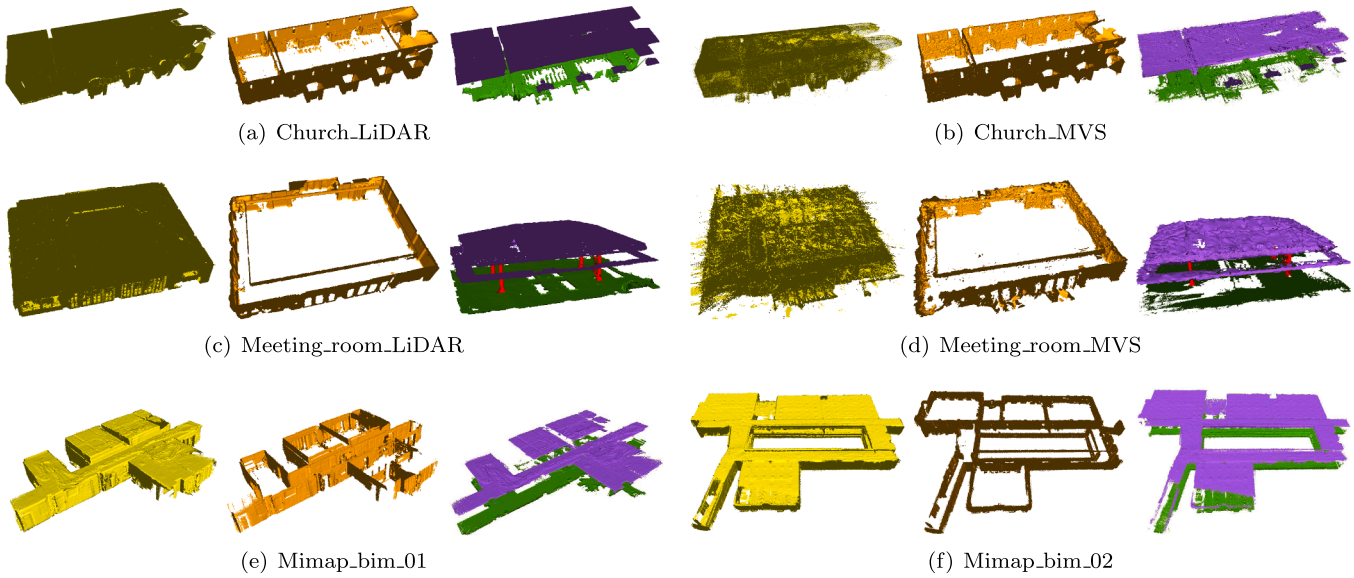


Fig. 13. Scene segmentation results on three datasets we used. For each scene, the three displays are input data, segmented wall point cloud and segmented non-wall point cloud (includes ceiling, floor and cylinder) respectively. Different semantic point cloud is shown with different colors. The segmentation results on Mimap_bim_00 and Office_MVS scenes are displayed in Fig. 3 and Fig. 4.

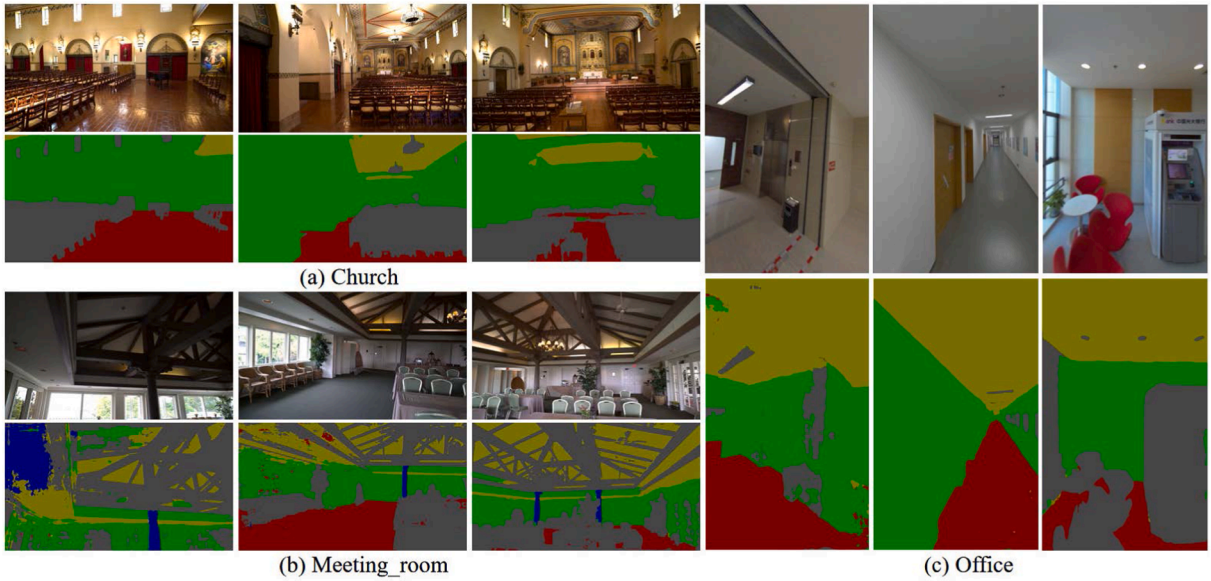


Fig. 14. Examples of 2D segmentation on MC and Office datasets. In (a), (b) and (c), the top row shows the input images and the bottom row is their segmentation results using fine-tuned DeepLabv3. The ceiling, floor, wall, cylinder and others are colored as yellow, red, green, blue and grey respectively.

Table 2
Key parameters of the proposed method.

Parameters and Values	Descriptions
$p_{min} = 5/10/100(*100)$	the least point number to support a plane
$d_{max} = 0.05$ m	the maximum distance between an inlier and a plane
$P_{miss} = 0.001$	the probability to miss the largest planes
$thre_{dev} = 0.85/0.90$	the maximum deviation of points in one plane
$thre_{ang} = 0° - 5°$	the threshold for regularizing planes
$thre_{ang2} = 15°$-$25°$	the angle threshold for merging close wall segments
$thre_{num} = 10$–20	the number threshold for merging close wall segments
$\lambda_1 = 0.5, \lambda_2 = 0.2, \lambda_3 = 0.3$	the balance parameters in the function for wall segment selection (Eq. (3))
$\beta_1 = 0.5, \beta_2 = 0.5$	the balance parameters in the function for wall structure reconstruction (Eq. (12))

planes, enhancing the regularity of the wall planes. Then, the threshold $thre_{ang2}$ and $thre_{num}$ are used to merge close wall segments to obtain a cleaner result. The higher the three parameters are, the more regular and cleaner the result is. Fig. 15 displays the results of different

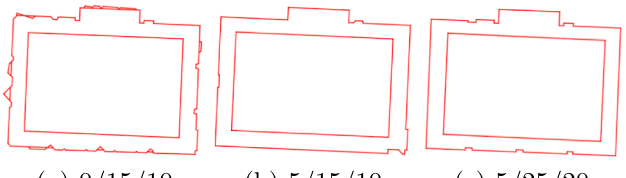


Fig. 15. Effect of different regularization parameters on Meeting_room_LiDAR scene. The displayed three values are $thre_{ang}$, $thre_{ang2}$ and $thre_{num}$ respectively.

regularization parameters. As can be seen in Fig. 15(a), there are more small structures without plane regularization, and larger values of $thre_{ang2}$ and $thre_{num}$ can lead to more regular results, such as the bottom right corner of the floorplan in (b) and (c). However, even with different regularization parameters, the floorplan is generally consistent, just with different degrees of detail and regularity. In our experiments, the results are acceptable when $thre_{ang}$ is set to 0°-5°, the $thre_{ang2}$ is set to 15°-25°, and the $thre_{num}$ is set to 10–20.*Wall segment selection weights.* We transform the floorplan generation into an energy minimization defined by Eq. (3), where the first two terms are essential to generate a reasonable result, and the last term M controls the complexity of the final floorplan. A higher weight λ_3 in Eq. (3) will lead to a floorplan with fewer small structures. Fig. 16 shows the different results by gradually increasing the complexity weight λ_3 with $\lambda_2 = 0.2$ and $\lambda_1 = 1 - \lambda_2 - \lambda_3$. As displayed in Fig. 16, different energy weights give a balance between the model completeness and model complexity, and they can be adjusted according to different datasets. In our experiments, we set $\lambda_1 = 0.5, \lambda_2 = 0.2$ and the $\lambda_3 = 0.3$ on all of the datasets.*Wall reconstruction weights.* Due to the noise and missing areas in the point cloud, as well as the observation that not all walls are connected from the floor to the ceiling, it is unreasonable to directly lift the edges of the floorplan to the height defined by the ceilings and floors. Considering the model closure and the balance between the point number and point coverage, we define the energy function Eq. (12) to select the final wall structures. Adjusting the values of β_1 and β_2 in Eq. (12) can give a different importance on the data completeness, as shown in Fig. 17. As can be seen in Fig. 17 consider only the point number term or point coverage term, obtaining the results that are not the best recovery of the point cloud (a). By contrast, (c) balances both terms and obtains a more reasonable model. In our experiments, we set β_1 and β_2 both to 0.5 on all of the datasets.

3.4. Reconstruction results evaluation

In this section, we evaluate the 3D reconstructed models of the proposed method on three datasets with different characteristics and complexity, and we compare them with four state-of-the-art methods. The first method is a general vectorized reconstruction method, named Polyfit (Nan and Wonka, 2017), which adopts the *slice-then-selection* strategy in 3D space, and the floorplan generation stage in our pipeline can be viewed as a similar 2D version of it. The remaining three are mesh decimation methods, including SAMD (Salinas et al., 2015), VSA (Cohen-Steiner et al., 2004) and QEM (Garland and Heckbert, 1997), which aim to simplify the dense mesh model to a lightweight model. Among the four state-of-the-art methods, the input of Polyfit is a point cloud while the remaining three methods take the mesh as input. Although the last three methods focus on reducing the facet number of the reconstructed dense meshes and the inputs are not exactly the same as with our method; however, our motivation and expected output are very similar, that is to obtain a compact model. Thus, the comparisons with these methods are meaningful. For LiDAR point cloud (including the MiMAP dataset and the Meeting_room_LiDAR and Church_LiDAR),

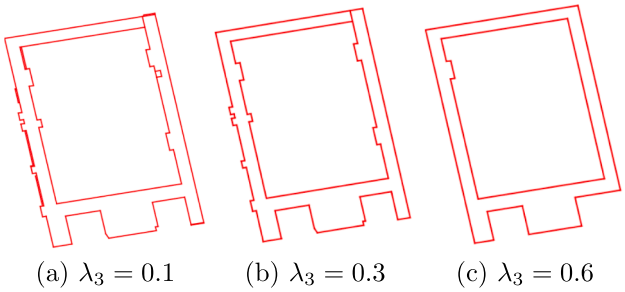


Fig. 16. Effect of different model complexity weight λ_3 in Eq. (3) on Mimap._bim_00 scene.

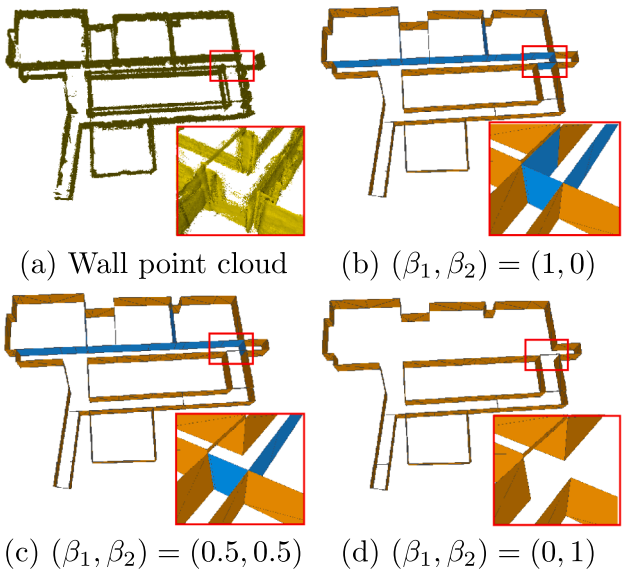


Fig. 17. Effect of different (β_1, β_2) in Eq. (12) on Mimap_bim_02 scene. In (b), (c) and (d), the orange walls are initial wall solution $Wall_i$ and the blue walls are optimal results after solving Eq. (12). The (c) balance the point number and point coverage, resulting in a more reasonable and closer wall model to (a) compared with (b) that only considers the point number term and (d) that only considers the poin.t coverage term.

we perform Poisson surface reconstruction on it to obtain the dense mesh. For MVS point cloud (including the Office dataset and the Meeting_room_MVS and Church_MVS), we obtain the mesh using visibility-based meshing implemented in OpenMVS (openMVS, 2020). Note that the reconstructed mesh using visibility-based meshing is more detailed than that using Poisson surface reconstruction because the former makes full use of the visibility information provided by calibrated cameras; however, it is suitable only for MVS point cloud. After meshing the point cloud, we take the generated meshes as input of SAMD, VSA and QEM. For Polyfit, we take the original LiDAR point cloud on the MiMAP dataset and the point cloud resampled uniformly from generated meshes on the MC and Office dataset as its input. Because the inputs of our reconstruction stage on the Meeting_room_LiDAR and Church_LiDAR scenes are the resampled point cloud from semantic mesh, we also take the resampled point cloud rather than the original LiDAR point cloud as the input for Polyfit on these two scenes to fairly evaluate the results between ours and Polyfit.

3.4.1. Evaluation on the MC dataset

The MC dataset contains two indoor scenes that, in total, include two LiDAR point cloud and two MVS point cloud. We adopt the *Segmentation-Fusion* strategy to obtain the segmented point cloud and then take the result as input to perform the reconstruction stage. Since there is no vectorized model ground truth on this dataset, we use the dense LiDAR point cloud with high precision as the ground truth and make the quantitative analysis using them. Fig. 18 displays the reconstruction results of our method and the other four methods.*Model correctness.* We resample 500 k points on the reconstructed model and compute the mean Hausdorff error from the point sets to the ground truth to measure the model correctness. Compared with LiDAR data, it is more challenging to reconstruct models with the MVS data since the latter usually inevitably contains more noise and missing areas. As seen in Fig. 18, our method can generate accurate and regular models with comparable error on the LiDAR input and the lowest error on the MVS input compared with the other four approaches. Polyfit depends on the quality of the detected plane primitives, and one missing plane may greatly reduce the model's quality, such as the result on Meeting_room_MVS in Fig. 18(b). By contrast, we decompose the 3D reconstruction into a set of

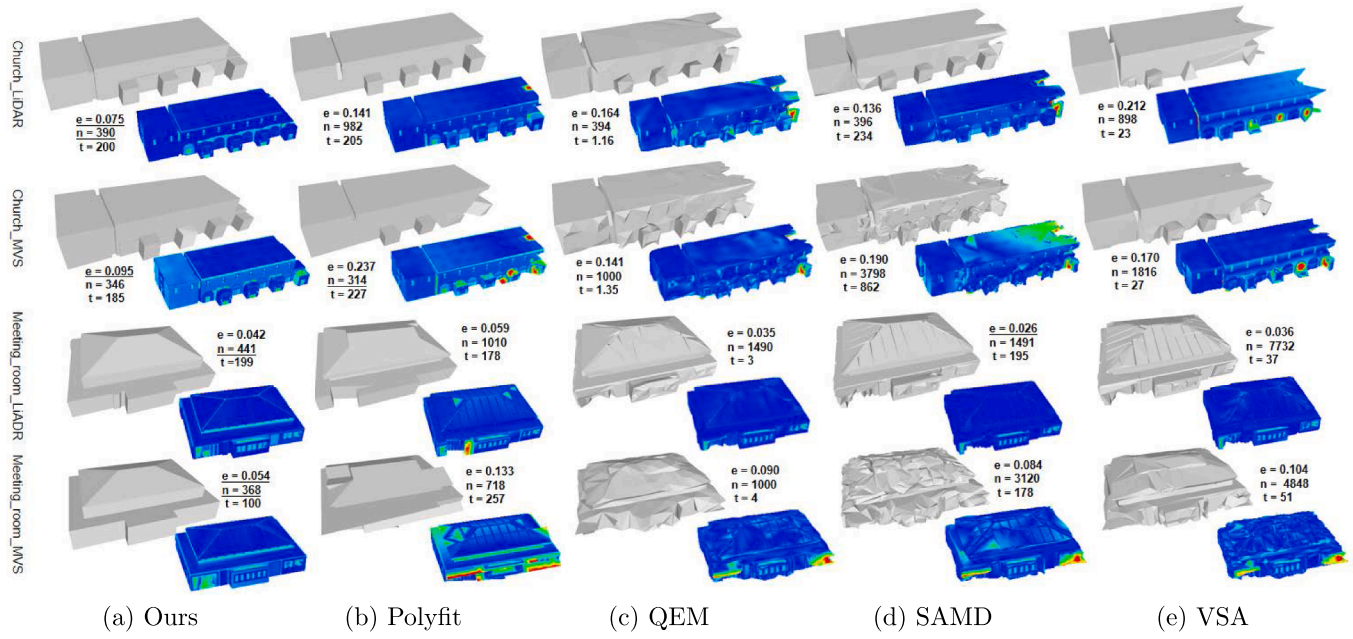


Fig. 18. Reconstruction results on the MC dataset for our method, Polyfit, QEM, SAMD and VSA. We compute and display the mean Hausdorff error(m) **e** from the output to LiDAR point cloud and show the error models with the meaning that the bluer the color, the smaller the error and the redder the color, the bigger the error. In addition, we also record the number of facets of the final model **n** and the reconstruction running time(sec) **t**. The lowest error and the minimum facet number on each dataset are underlined.

2D optimizations, and the wall, ceiling and floor structures are detected and reconstructed in sequence, which reduces the impact of the noise and missing areas and can generate more robust and faithful results. In addition, the ideal output of our motivation is the vectorized compact model with high regularity and high abstractness, which is not the exactly the same with the input point cloud or meshes. The SAMD, VSA and QEM simplify the meshes while retaining the original shape as much as possible, which makes their results more dependent on the input models and lack of regularity. Compared with these three methods, we can obtain accurate models with high regularity even on the defect-laden MVS data.*Model simplicity*. The number of facets in the final models are counted and listed in Fig. 18. Our method and Polyfit aim to directly generate the compact polygonal models with piecewise planes from the input and can greatly reduce the model redundance. Compared with this strategy, the SAMD, VSA and QEM obtain their simplified models by reducing the facet number on dense meshes, which does not make it easy to achieve the same model simplicity as ours while keeping good reconstruction results. As seen in Fig. 13, unlike the LiDAR point cloud ((a), (c)), the MVS point cloud ((b), (d)) contains more noise and outliers, making most of the results worse on the MVS data. Compared with other methods, our approach not only obtains compact models with fewer facets (on average, 20% of the VSA, 37% of the SAMD, 50% of the QEM and 59% of the Polyfit), but also generates more stable results on both the MVS and LiDAR point cloud. For the proposed method, more details could be obtained on the LiDAR data than on the MVS data (more facets on LiDAR than MVS, 390 vs. 346 on Church, and 441 vs. 368 on Meeting_room), but their reconstruction accuracy was comparable (the model error on LiDAR and MVS is 0.075 vs. 0.095 on Church, and 0.042 vs. 0.054 on Meeting_room).*Efficiency*. Considering that the semantic segmentation methods in our pipeline are replaceable and a small number of rough segmentation results are acceptable, the segmentation part can be viewed as a preprocessing stage, and we mainly compare the reconstruction time of our method with other state-of-the-art methods. We record the reconstruction running time of our method and others in Fig. 18 and list the running time of each step of our pipeline in Table 1. The QEM iteratively contracts vertex pairs using quadric matrices, and the VSA partitions the mesh and approximates the shape with geometric proxies. Both of them perform quickly while having low regularity. The SAMD collapses edges with the help of planar proxies added into the local error metric, and the running time is longer than VSA and QEM. Our method and Polyfit have a more similar motivation, and their running time is comparable. Polyfit performs the *slice-then-selection* in 3D space, and the number of detected plane primitives is crucial to balancing the reconstruction accuracy and the running time. By contrast, our method can alleviate the impact of the plane number with the dimension reduction operation.

In addition, we show the internal reconstruction results of the proposed method in Fig. 19. Compared with other indoor reconstruction methods that can only address horizontal ceilings and floors (Ochmann et al., 2016; Ochmann et al., 2019; Cui et al., 2019; Wang et al., 2020, our approach can recover slant ceiling and floor structures (*e.g.* the slant roofs in Fig. 19(a)(b), and the step on the floor in Fig. 19)) and can obtain vectorized LoD2 models without using the Manhattan or Atlanta world assumptions, enhancing the generality of the proposed method for indoor scenes with different characteristics and complexity.

3.4.2. Evaluation on the MiMAP dataset

The MiMAP dataset includes three LiDAR point cloud with various complexity and the corresponding BIM line framework ground truth. We manually draw the 3D models according to the line frameworks, as shown in Fig. 20, and take them as the vectorized reconstruction ground truth on which we compare the model correctness of different methods. In addition, we compute the mean Hausdorff error from the generated models to the LiDAR point cloud to measure the difference between the output and the original input. The reconstruction results and comparisons are shown in Fig. 21.*Model correctness.* As seen in Fig. 21, our method can obtain the vectorized models with the lowest error on all of the three datasets except for Mimap_bim_02, where the point cloud error e_p of ours is slightly higher than that of Polyfit, while the ground truth error e_g is greatly lower. Since we measure the correctness with the mean Hausdorff error from the output to the ground truth and the input point cloud, the missing areas on the LiDAR point cloud, as shown in Fig. 22, will influence the error computation, resulting in a slightly higher error than that on the MC dataset. In addition, the point cloud has some

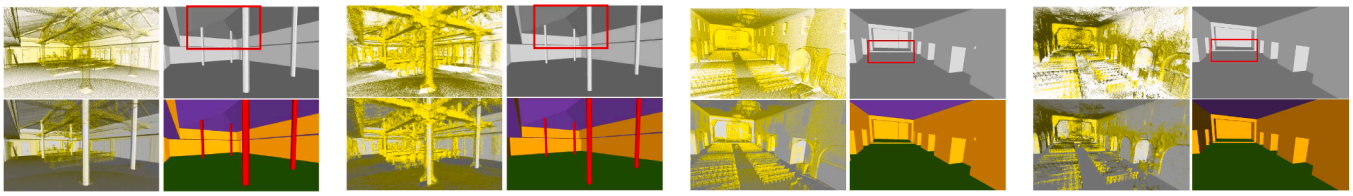


Fig. 19. Internal reconstruction results on the MC dataset. For each scene, the first row displays the dense point cloud and our reconstruction result. The second row displays the result by putting the point cloud and our result together, and the reconstruction result with semantic information. Without the assumption that ceilings and floors consist of horizontal planes, we can recover non-horizontal structures in ceilings and floors, such as the slant roofs in (a) and (b), and the step on the floor in (c) and (d) (which are circled with red rectangles).

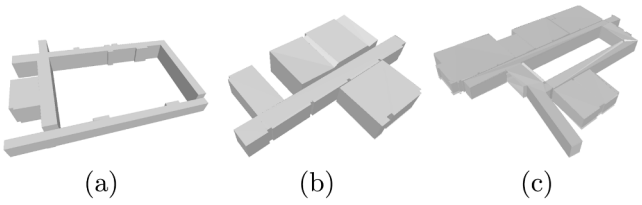


Fig. 20. Vectorized reconstruction ground truth on the MiMAP dataset. The (a)-(c) are the ground truths of Mimap_bim_00-02.

ambiguous parts, and the BIM line framework ground truths do not totally obey the point cloud, which also affects the error computation, as shown in Fig. 22(a), where the ceiling contour in the BIM line framework is slightly higher than most of the ceiling points. The mesh simplification methods, including SAMD, VSA and QEM, can obtain models as close to the input as possible, and thus, their errors on the LiDAR point cloud are lower than those on the vectorized ground truths. However, as seen in Fig. 21, even on the LiDAR point cloud, the three mesh simplification methods still generate many defects and errors that are obviously larger than ours.*Model simplicity and Efficiency*. As shown in Fig. 21, our method can obtain enough compact models with high simplicity in a reasonable running time. Polyfit takes a very long time to obtain the final model on Mimap_bim_01 and Mimap_bim_02 because it intersects and selects planes in 3D space and cannot handle input with too many plane primitives. Compared with this approach, we transform the 3D reconstruction into a sequence of 2D optimizations, which reduces the reconstruction complexity and enhances the scalability of our method.

3.4.3. *Evaluation on the Office dataset*

The Office dataset includes a dense MVS point cloud that is reconstructed by COLMAP (Schonberger and Frahm, 2016). This scene contains masses of weak texture regions, such as white walls and glass walls, which inevitably lead to a large area that is missing, and a raft of noise in the MVS point cloud, as shown in Fig. 23. This problem can be alleviated by transforming the MVS point cloud into a triangular mesh. However, the mesh also has many areas of noise and a large hole that cannot be repaired, as shown in Fig. 23(b), where the large hole is circled with a red rectangle.

Due to the lack of ground truth, we mainly compare the reconstruction results qualitatively on the dataset, as shown in Fig. 24, where we also record the facet number of the final models and the running time. As can be seen in Fig. 24, the proposed method can generate an accurate and complete model even though the MVS data has obvious noise and missing parts. However, the SAMD, VSA and QEM can only simplify the meshes and thus cannot address the obvious missing parts. Polyfit can resist certain missing data; however, some small structures, such as the elevators, could be missing, as shown in Fig. 24(c) with the red bounding box. By contrast, with the label consistency adjustment

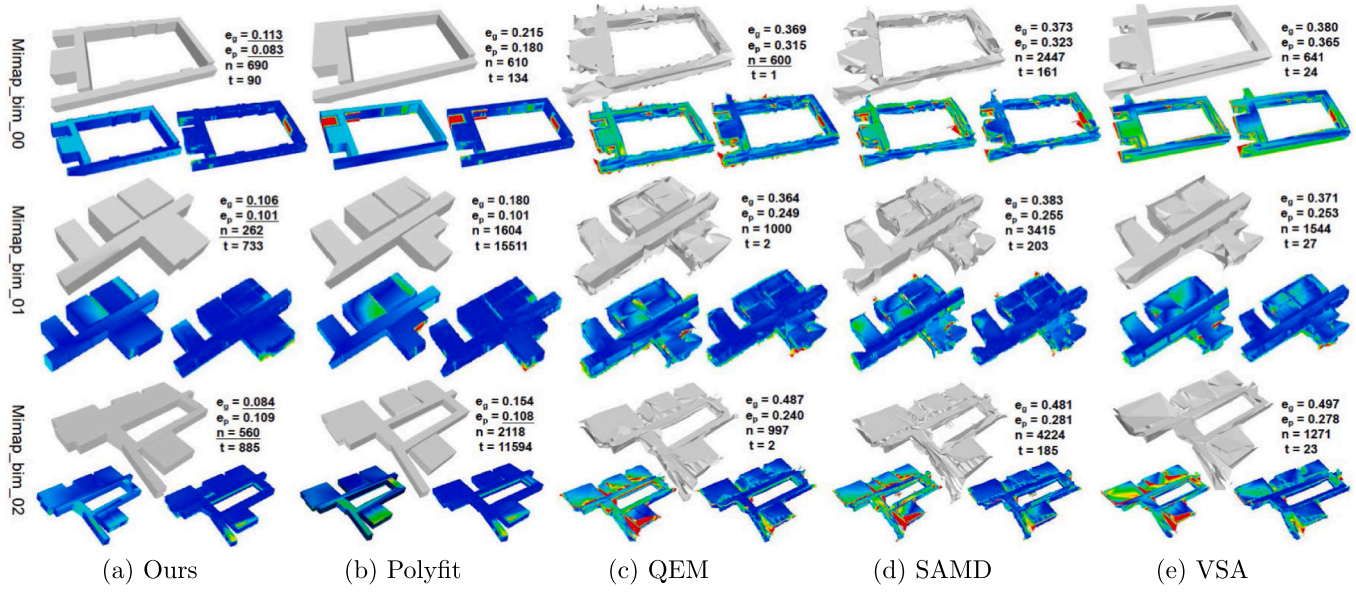


Fig. 21. Reconstruction results on the MiMAP dataset for our method, Polyfit, QEM, SAMD and VSA. For each reconstruction result, the mean Hausdorff error(m) e_g and the left error model are computed from the output to the vectorized ground truth. And the mean Hausdorff error(m) e_p and the right error model are computed from the output to the LiDAR point cloud. In addition, the number of facets of the final model **n** and the reconstruction running time(sec) **t** are recorded. The lowest error and the minimum facet number on each dataset are underlined.

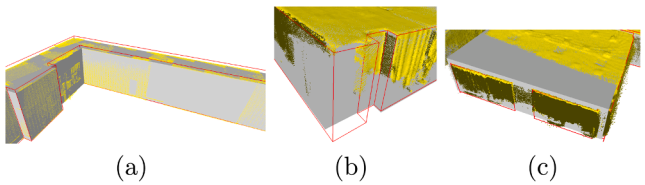
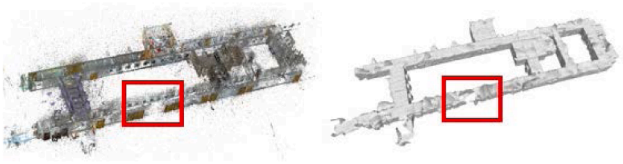
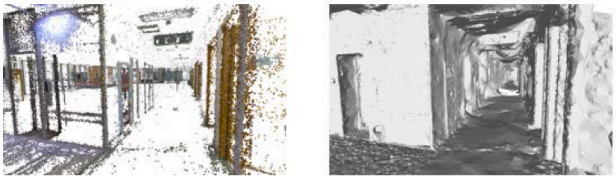


Fig. 22. Error analyses on the MiMAP dataset. (a)-(c) are three close-ups on Mimap_bim_00-02 respectively. In each close-up, we display the Lidar point cloud (yellow), our reconstruction result (grey) and the BIM line framework ground truth (red). In (a) and (b), the wall part in LiDAR point cloud is obvious missing and in (c) the ceiling part exists the missing areas, which affect the computation of error e_p. Besides, we can see that in (a), the ceiling contour of the BIM line framework is slightly higher than most of the ceiling points in LiDAR point cloud, which affects the computation of error e_g.

(a) Example images of Office_MVS scene



(b) MVS point cloud (left) and MVS mesh (right)



(c) Close-up of point cloud (left) and mesh (right)

Fig. 23. Data presentation on the Office dataset. The Office_MVS scene contains lots of weak texture areas (*e.g.* white walls and glass walls in (a)), leading to many missing areas and noise in the MVS point cloud and MVS mesh (*e.g.* the area circled with red rectangle in (b) and the close-up in (c)), which make it more difficult to obtain a complete and accurate vectorized model.

following Eq. (11) and the global optimization following Eq. (12), the proposed method can handle a degree of noise and missing areas and thus is more robust in a variety of indoor scenes.

3.5. Comparison with learning-based reconstructions

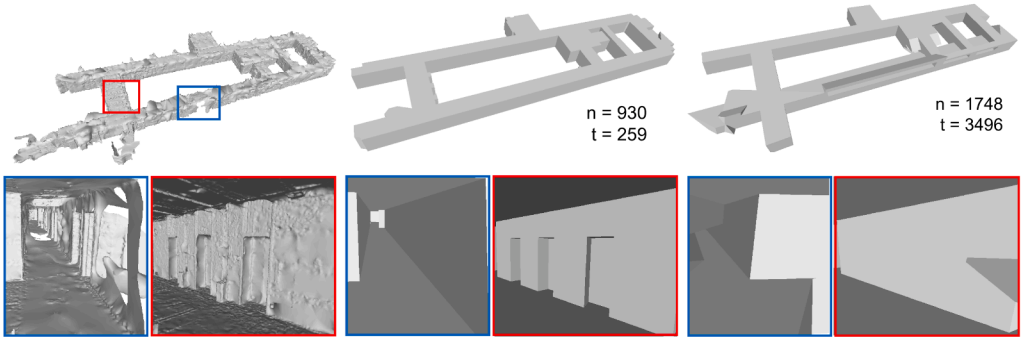
In recent years, with the rapid development of deep learning, much work has used neural networks to help solve the indoor reconstruction problem. Many of these studies used the network to extract corners and room segments from point cloud or images, and then incorporated them into traditional optimizations to obtain the final model (Phalak et al., 2020; Liu et al., 2018; Chen et al., 2019). Recently, the method in (Liu et al., 2021) used an end-to-end network architecture to directly convert 3D points into models, but this approach is usually suitable only for small objects and relatively complete point clouds. In this section, we compare the proposed method with two state-of-the-art and open-sourced learning-based reconstruction methods, FloorNet (Liu et al., 2018) and Floor-SP (Chen et al., 2019). FloorNet takes the Manhattan-rectified point cloud and images (optional) as input and combines three network branches and Integer Programming to obtain the floor-plan. Floor-SP takes the point-density/normal image from a top-down view as input, uses the network to obtain room segmentation results and corners/edges likelihood, and defines the energy function, followed by solving it with sequentially finding room-wise shortest paths, which can handle non-Manhattan scenes. Note that our method generates 3D vectorized models, including ceiling, floor, wall and cylinder structures; however, the above two methods focus on the floorplan modeling. Thus, we ortho-project our reconstructed wall models onto the ground to generate floorplans and use them to compare our method with the two methods. In experiments, we used the segmented wall point cloud on three datasets as the input of FloorNet, and the top-down density/normal map of the wall point cloud as the input of Floor-SP. Fig. 25 displays the floorplan results of ours and these two methods.

As can be seen in Fig. 25, compared with the other two learning-based methods, the results of our method are more consistent with the input point cloud, and are more complete and regular. FloorNet and Floor-SP rely on relatively good semantic segmentation results as the input for subsequent optimization. Because the datasets used in the paper and the training data used in FloorNet and Floor-SP are not completely the same in structure, the semantic segmentation results on some of the datasets are poor, and eventually, the floorplans are unsatisfactory, such as the results in Fig. 25 (e), (h). Thus, the generalization of the two methods is limited, and training on larger datasets may alleviate this problem. In addition, the two methods use a small resolution map (256*256) which makes it difficult to capture structural details, reducing the quality of the final results. In terms of the running time, FloorNet is fast and takes an average of 13.8s on the three datasets. Floor-SP is slower and may take a long time to find the shortest path in the optimization stage, and the average time is 607.2s on the three datasets. Different from these two methods, we take advantage of the geometric information of the point cloud and the verticality of the facades to transform the reconstruction into 2D space, and we obtain the floorplan by solving the optimization problem, which has good robustness to scenes with different structural complexity, noise and missing areas, and the average time is 331.5s on the three datasets. However, using the network to extract high-level features and make predictions and then integrating inferred information with geometric optimization methods is worthy of in-depth thinking when solving reconstruction problems, which is also our concern.

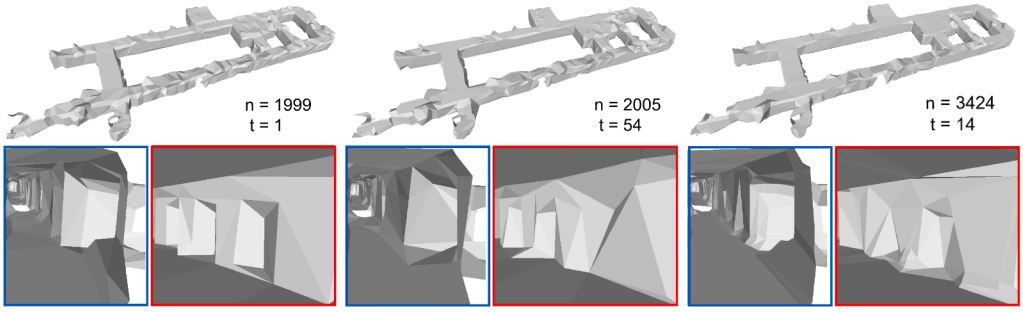
4. Conclusions

In this paper, we propose a complete and effective multistep pipeline for reconstructing indoor scenes with a vectorized representation that confirms to the CityGML 3.0 (Kutzner et al., 2020) LoD2 and without the Manhattan or Atlanta world assumptions. Different from the general strategy in that the reconstruction is performed as a whole in 3D space, we decompose the 3D reconstruction into a sequence of 2D segment or cell assembly problems by means of the semantic information, and the final results can be obtained by solving each global optimization sub-problem and then combining the respective results. The proposed strategy reduces the reconstruction complexity and has the ability to restore complex indoor permanent structures including sloping ceilings and floors, and is robust to different styles of the scenes. Experiments show that our method can generate accurate models with high simplicity and semantic information on both precise LiDAR data and defect-laden MVS data, which also demonstrates the generality of the proposed method.

In our approach, the wall planes are detected using RANSAC. However, some small planes could be missing if there are serious deficiencies in the point cloud, which will affect the reconstruction results, such as

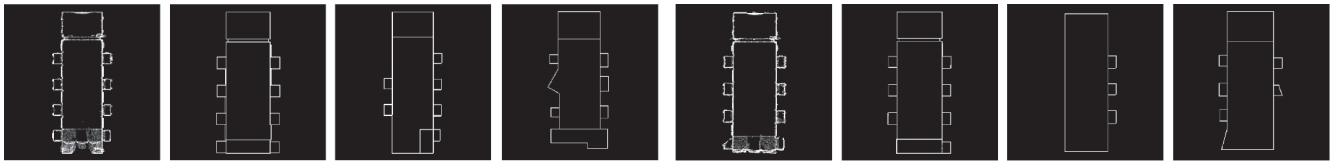


(a) Dense MVS mesh (b) Ours (c) Polyfit

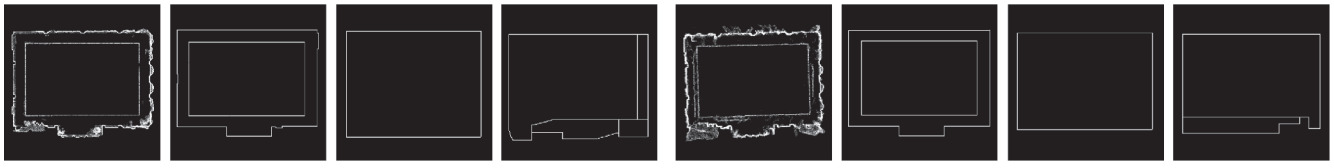


(d) QEM (e) SAMD (f) VSA

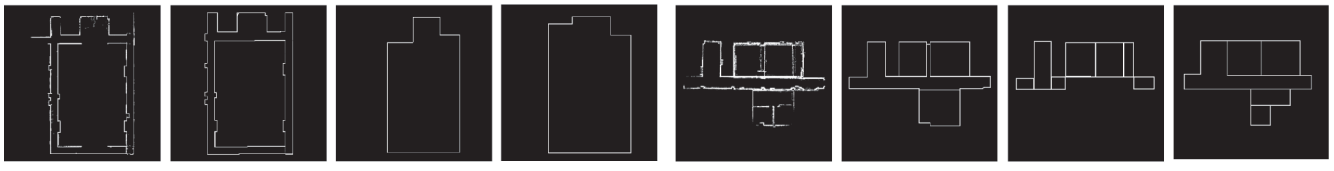
Fig. 24. Reconstruction results on the Office dataset for our method, Polyfit, QEM, SAMD and VSA. For each subpart, the bottom row displays two close-ups. The number of facets of the final model **n** and the reconstruction running time(*sec*) **t** are also recorded.



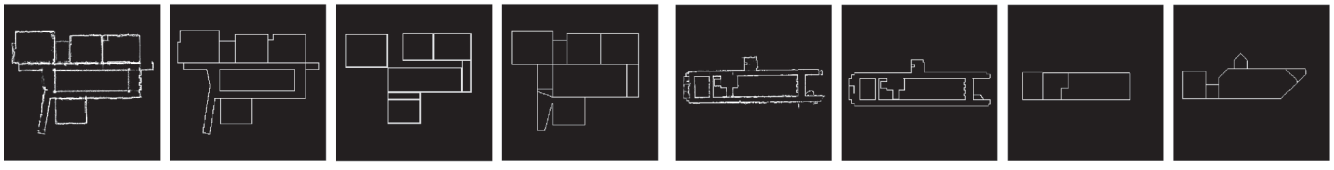
(a) Church LIDAR (b) Church MVS



(c) Meeting room LIDAR (d) Meeting room MVS



(e) Mimap bim 00 (f) Mimap bim 01



(g) Mimap bim 02 (h) Office MVS

Fig. 25. 2D floorplan reconstruction results of ours and learning-based methods on evaluated datasets. In (a)-(h), from left to right, we display the density map of the point cloud, the result of our method, FloorNet and Floor-SP, respectively.

the protruding facade in Fig. 22(b). Consequently, in the future, we will aim to enhance the plane detection and floorplan generation using other available information such as images for MVS data. In addition, for generating the vectorized models closer to the real scenes, we will focus on detailing our LoD2 model to obtain a model with richer structures (*e. g.* LoD3).

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Bauchet, J.P., Lafarge, F., 2020. Kinetic shape reconstruction. ACM Transactions on Graphics 39 (5), 1–14.

Becker, S., Peter, M., Fritsch, D.. Grammar-supported 3d indoor reconstruction from point clouds for "as-built" bim. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences 2015;II-3/W4:17–24.

Berger, M., Tagliasacchi, A., Seversky, L.M., Alliez, P., Guennebaud, G., Levine, J.A., Sharf, A., Silva, C.T., 2017. A survey of surface reconstruction from point clouds. Computer Graphics Forum 36 (1), 301–329.

Billen, R., Zaki, C.E., Servières, M., Moreau, G., Hallot, P. Developing an ontology of space: Application to 3d city modeling. In: Usage, Usability, and Utility of 3D City Models – European COST Action TU0801. 2012. p. 02007.

Bouzas, V., Ledoux, H., Nan, L., 2020. Structure-aware building mesh polygonization. ISPRS Journal of Photogrammetry and Remote Sensing 167, 432–442.

Boykov, Y., Kolmogorov, V., 2004. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE transactions on pattern analysis and machine intelligence 26 (9), 1124–1137.

Boykov, Y., Veksler, O., Zabih, R., 2001. Fast approximate energy minimization via graph cuts. IEEE Transactions on pattern analysis and machine intelligence 23 (11), 1222–1239.

Chen, J., Liu, C., Wu, J., Furukawa, Y. Floor-sp, 2019. Inverse cad for floorplans by sequential room-wise shortest path. In: In: International Conference on Computer Vision.

Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H., 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In: European conference on computer vision, pp. 801–818.

Cohen-Steiner, D., Alliez, P., Desbrun, M., 2004. Variational shape approximation. ACM Transactions on Graphics 23 (3), 905–914.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B. The cityscapes dataset for semantic urban scene understanding. https://www.cityscapes-dataset.com; 2016.

Cui, H., Gao, X., Shen, S., Hu, Z., 2017. HSfM: Hybrid structure-from-motion. IEEE Conference on Computer Vision and Pattern Recognition. 2393–2402.

Cui, Y., Li, Q., Yang, B., Xiao, W., Chen, C., Dong, Z., 2019. Automatic 3-d reconstruction of indoor environment with mobile laser scanning point clouds. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 12 (8), 3117–3130.

Gamrath, G., Anderson, D., Bestuzheva, K., Chen, W.K., Eifler, L., Gasse, M., Gemander, P., Gleixner, A., Gottwald, L., Halbig, K., et al. The SCIP Optimization Suite 7.0. Technical Report; Optimization Online; 2020. URL http://www.optimization-online.org/DB_HTML/2020/03/7705.html.

Garland, M., Heckbert, P.S. Surface simplification using quadric error metrics. In: ACM SIGGRAPH Computer Graphics. 1997. p. 209–216.

Horna, S., Meneveaux, D., Damiand, G., Bertrand, Y., 2009. Consistency constraints and 3d building reconstruction. Computer-Aided Design 41 (1), 13–27.

Ikehata, S., Yang, H., Furukawa, Y., 2015. Structured indoor modeling. International Conference on Computer Vision. 1323–1331.

Kaiser, A., Ybanez Zepeda, J.A., Boubekeur, T., 2019. A survey of simple geometric primitives detection methods for captured 3d data. Computer Graphics Forum 38 (1), 167–196.

Kang, Z., Yang, J., Yang, Z., Cheng, S., 2020. A review of techniques for 3d reconstruction of indoor environments. International Journal of Geo-Information 9, 1–31.

Kazhdan, M., Bolitho, M., Hoppe, H., 2006. Poisson surface reconstruction. In: Eurographics Symposium on Geometry Processing., pp. 61–70.

Knapitsch, A., Park, J., Zhou, Q.Y., Koltun, V., 2017. Tanks and temples: Benchmarking large-scale scene reconstruction. ACM Transactions on Graphics 36 (4), 1–13.

Kutzner, T., Chaturvedi, K., Kolbe, T.H., 2020. Citygml 3.0: New functions open up new applications. PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation. Science 88 (1), 43–61.

Lee, S., Feng, D., Grimm, C., Gooch, B., 2008. A sketch-based user interface for reconstructing architectural drawings. Computer Graphics Forum 27 (1), 81–90.

Li, M., Nan, L., 2021. Feature-preserving 3d mesh simplification for urban buildings. ISPRS Journal of Photogrammetry and Remote Sensing 173, 135–150.

Li, M., Rottensteiner, F., Heipke, C., 2019. Modelling of buildings from aerial lidar point clouds using tins and label maps. ISPRS Journal of Photogrammetry and Remote Sensing 154, 127–138.

Li, T., Shu, B., Qiu, X., Wang, Z., 2010. Efficient reconstruction from architectural drawings. International journal of computer applications in technology 38 (1–3), 177–184.

Liu, C., Wu, J., Furukawa, Y. Floornet, 2018. A unified framework for floorplan reconstruction from 3d scans. In: European Conference on Computer Vision..

Liu, Y., D'Aronco, S., Schindler, K., Wegner, J.D. Pc2wf: 3d wireframe reconstruction from raw point clouds. In: International Conference on Learning Representations. 2021.

Mura, C., Mattausch, O., Pajarola, R., 2016. Piecewise-planar reconstruction of multi-room interiors with arbitrary wall arrangements. Computer Graphics Forum 35, 179–188.

Musialski, P., Wonka, P., Aliaga, D.G., Wimmer, M., Gool, L., Purgathofer, W., 2013. A survey of urban reconstruction. Computer Graphics Forum 32 (6), 146–177.

Nan, L., Wonka, P. Polyfit, 2017. Polygonal surface reconstruction from point clouds. International Conference on Computer Vision. 2353–2361.

Ochmann, S., Vock, R., Klein, R., 2019. Automatic reconstruction of fully volumetric 3d building models from oriented point clouds. ISPRS Journal of Photogrammetry and Remote Sensing 151, 251–262.

Ochmann, S., Vock, R., Wessel, R., Klein, R., 2016. Automatic reconstruction of parametric building models from indoor point clouds. Computers & Graphics 54, 94–103.

Oesau, S., Lafarge, F., Alliez, P., 2014. Indoor scene reconstruction using feature sensitive primitive extraction and graph-cut. ISPRS Journal of Photogrammetry and Remote Sensing 90, 68–82.

openMVS, https://github.com/cdcseacave/openMVS.

Phalak, A., Badrinarayanan, V., Rabinovich, A. Scan2plan: Efficient floorplan generation from 3d scans of indoor scenes. arXiv preprint arXiv:200307356 2020;.

Pintore, G., Mura, C., Ganovelli, F., Fuentes-Perez, L., Pajarola, R., Gobbetti, E., 2020. State-of-the-art in automatic 3d reconstruction of structured indoor environments. Computer Graphics Forum 39 (2), 667–699.

Previtali, M., Díaz Vilariño, L., Scaioni, M., 2018. Indoor building reconstruction from occluded point clouds using graph-cut and ray-tracing. Applied Sciences 8, 1529.

Previtali, M., Scaioni, M., Barazzetti, L., Brumana, R.. A flexible methodology for outdoor/indoor building reconstruction from occluded point clouds. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences 2014;II-3: 119–126.

Qi, C.R., Su, H., Mo, K., Guibas, L.J. Pointnet, 2017a. Deep learning on point sets for 3d classification and segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition., pp. 652–660.

Qi, C.R., Yi, L., Su, H., Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: International Conference on Neural Information Processing Systems. 2017b. p. 5105–5114.

Salinas, D., Lafarge, F., Alliez, P., 2015. Structure-aware mesh decimation. Computer Graphics Forum 34 (6), 211–227.

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

Schonberger, J.L., Frahm, J.M., 2016. Structure-from-motion revisited. In: IEEE Conference on Computer Vision and Pattern Recognition., pp. 4104–4113.

Schönberger, J.L., Zheng, E., Pollefeys, M., Frahm, J.M., 2016. Pixelwise view selection for unstructured multi-view stereo. In: European Conference on Computer Vision., pp. 501–518.

Shen, S., 2013. Accurate multiple view 3d reconstruction using patch-based stereo for large-scale scenes. IEEE transactions on image processing 22 (5), 1901–1914.

Sun, C., Hsiao, C.W., Sun, M., Chen, H.T. Horizonnet, 2019. Learning room layout with 1d representation and pano stretch data augmentation. In: IEEE Conference on Computer Vision and Pattern Recognition., pp. 1047–1056.

The CGAL Project,. The computational geometry algorithms library. https://www.cgal.org; 2019.

Thomson, C., Boehm, J., 2015. Automatic geometry generation from point clouds for BIM. Remote Sensing 7 (9), 11753–11775.

Tran, H., Khoshelham, K.. A stochastic approach to automated reconstruction of 3d models of interior spaces from point clouds. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences 2019;IV-2/W5:299–306.

Tran, H., Khoshelham, K., 2020. Procedural reconstruction of 3d indoor models from lidar data using reversible jump markov chain monte carlo. Remote Sensing 12, 838.

Tran, H., Khoshelham, K., Kealy, A., Díaz Vilariño, L., 2018. Shape grammar approach to 3d modeling of indoor environments using point clouds. Journal of Computing in Civil Engineering 33 (1), 14.

vanDongen, S. A cluster algorithm for graphs. Technical Report R 0010; 2000.

Vu, H.H., Labatut, P., Pons, J.P., Keriven, R., 2011. High accuracy and visibility-consistent dense multiview stereo. IEEE transactions on pattern analysis and machine intelligence 34 (5), 889–901.

Wang, C., Hou, S., Wen, C., Gong, Z., Li, Q., Sun, X., Li, J., 2018. Semantic line framework-based indoor building modeling using backpacked laser scanning point cloud. ISPRS journal of photogrammetry and remote sensing 143, 150–166.

Wang, R., Xie, L., Chen, D., 2017. Modeling indoor spaces using decomposition and reconstruction of structural elements. Photogrammetric Engineering and Remote Sensing 83, 827–841.

Wang, S., Cai, G., Cheng, M., Junior, J.M., Huang, S., Wang, Z., Su, S., Li, J., 2020. Robust 3d reconstruction of building surfaces from point clouds based on structural and closed constraints. ISPRS Journal of Photogrammetry and Remote Sensing 170, 29–44.

Wen, C., Dai, Y., Xia, Y., Lian, Y., Tan, J., Wang, C., Li, J., 2020. Toward efficient 3-d colored mapping in gps-/gnss-denied environments. IEEE Geoscience and Remote Sensing Letters 17 (1), 147–151.

Xia, S., Chen, D., Wang, R., Li, J., Zhang, X., 2020. Geometric primitives in lidar point clouds: A review. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 13, 685–707.
Yang, S.T., Wang, F.E., Peng, C.H., Wonka, P., Sun, M., Chu, H.K. Dula-net, 2019. A dual-projection network for estimating room layouts from a single rgb panorama. In: IEEE Conference on Computer Vision and Pattern Recognition., pp. 3363–3372.
Yu, D., Ji, S., Liu, J., Wei, S., 2021. Automatic 3d building reconstruction from multi-view aerial images with deep learning. ISPRS Journal of Photogrammetry and Remote Sensing 171, 155–170.
Zeng, W., Karaoglu, S., Gevers, T., 2020. Joint 3d layout and depth prediction from a single indoor panorama image. European Conference on Computer Vision. 666–682.
Zhou, Y., Shen, S., Hu, Z., 2018. Fine-level semantic labeling of large-scale 3d model by active learning. In: International Conference on 3D Vision, pp. 523–532.