ISPRS Journal of Photogrammetry and Remote Sensing 196 (2023) 490–501

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

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

FloorUSG: Indoor floorplan reconstruction by unifying 2D semantics and 3D geometry☆

Jiali Han [a,b,c,1], Yuzhou Liu [a,b,c,1], Mengqi Rong [a,b,c], Xianwei Zheng [d], Shuhan Shen [a,b,c,*]

[a] Institute of Automation, Chinese Academy of Sciences, Beijing, China
[b] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
[c] CASIA-SenseTime Research Group, Beijing, China
[d] The State Key Lab. LIESMARS, Wuhan University, Wuhan, China

ARTICLE INFO

Keywords:
Floorplan reconstruction
Plane detection
Integer programming

ABSTRACT

This paper proposes a multistage floorplan reconstruction approach from RGB images and a dense 3D mesh, called FloorUSG, by combining 2D plane instances and 3D plane primitives. In primitive detection, plane instances inferred from images complement the results of traditional 3D plane detection well. And in the optimization, both the plane confidence and the geometric quality of data are considered to select the optimal subset from the candidates. Different from existing methods that rely on delicate corner detection from a planar graph or pure geometric 3D plane detection, our framework accurately recovers the location of the floorplan via 2D–3D primitive fusion. Experimental results indicate that our method has the ability to recover detailed structures of scenes of different scales and can reconstruct the floorplan from imperfect data with high robustness compared to the state-of-the-art algorithms.

1. Introduction

A floorplan reflects the overall layout of the indoor facade structure inside a 3D building. And the floorplan reconstruction is an important research area in the field of computer vision and photogrammetry due to its great potential in robot localization (Boniardi et al., 2017; Wang et al., 2019), indoor scene understanding (Ziran and Marinai, 2018; Pintore et al., 2020b), reproduction (Liu et al., 2015) and so on. However, high-quality floorplan generation in industry is labor-intensive, and fully automated floorplan reconstruction is an urgent need. In this task, the core challenge lies in recovering an accurate and complete floorplan from a building interior with a complex structure and unavoidable occlusion.

Single panoramic images and point clouds are the two most common sources for floorplan reconstruction. When a panoramic image is used as input, floorplan generation is often converted into the detection of boundaries and some methods (Yang et al., 2019; Pintore et al., 2020a; Sun et al., 2019) infer the floorplan via end-to-end networks. These methods are quick at inference but are difficult to scale to large-scale scenes such as shopping malls. When inputting point clouds, the *detection-then-selection* strategy is preferred. The selection stage is always defined as an optimization problem and the difference between methods mainly lies in the detection phase. Some work (Han et al., 2021; Fang et al., 2021) detects and regularizes primitives from point clouds while other work (Liu et al., 2018; Chen et al., 2019; Stekovic et al., 2021) uses the network to infer primitives from the point density map. However, due to the interference of data acquisition equipment, illumination, and weak scene textures, the point cloud inevitably contains noise and missing areas, which may affect the robustness of the methods. In this paper, we aim to reconstruct floorplans from indoor scenes with different scales, and thus, the point cloud is more suitable to be adopted than a single panorama. For indoor scenes, the point cloud is usually obtained by mobile scanning using RGB-D devices such as Kinect or image-based reconstruction using SfM and MVS. Either way, the calibrated images (including camera intrinsics and poses) can be obtained and the dense mesh is available after meshing the point cloud.

This paper takes calibrated images and a dense mesh as input and integrates planes inferred from images with planes detected from 3D data into a unified regularization and optimization framework. First,

☆ This work was supported by the National Natural Science Foundation of China (No. U22B2055 and 62273345), and by the Beijing Natural Science Foundation (No. L223003).

* Corresponding author at: Institute of Automation, Chinese Academy of Sciences, Beijing, China.
E-mail addresses: jiali.han@nlpr.ia.ac.cn (J. Han), liuyuzhou2021@ia.ac.cn (Y. Liu), mengqi.rong@nlpr.ia.ac.cn (M. Rong), zhengxw@whu.edu.cn (X. Zheng), shshen@nlpr.ia.ac.cn (S. Shen).

[1] The first two authors, Jiali Han and Yuzhou Liu, contributed equally to this paper.

https://doi.org/10.1016/j.isprsjprs.2023.01.020
Received 22 June 2022; Received in revised form 20 January 2023; Accepted 25 January 2023

we use an off-the-shelf semantic segmentation network to infer images and segment indoor facades from point cloud sampling from the mesh. The labels of the point cloud are obtained by projecting points onto their visible images and then max-voting. Note that we take the mesh as input is because the mesh can disambiguate the visibility of points. Then, we detect plane primitives from indoor facade point clouds as a whole and locally using 2D plane instances inferred from a plane detection network. Finally, the floorplan is obtained through global optimization by considering both the plane confidence of points and the quality of the point cloud.

The main contributions of our work include the following:

- We propose a multilevel plane detection solution combining the semantics of images and the geometry of point clouds, which captures more detailed structures and enhances the robustness of traditional plane detection methods.
- We cast the floorplan reconstruction as a global optimization balancing the quality of the point cloud with the plane confidence of points derived from images.
- We propose an effective pipeline to generate floorplans from indoor scenes with high robustness to imperfect input data and inferences from the network by embedding the plane semantics inferred through the network into the reconstruction process based on geometric optimization.

2. Related work

The study of indoor floorplan reconstruction can be roughly divided into three categories: classical techniques, end-to-end networks, and hybrid schemes. Here, we briefly review each category.

Classical techniques. Some floorplan generation methods (Cabral and Furukawa, 2014; Pintore et al., 2018) rely on basic image processing techniques and point cloud reconstruction processes (Kangni and Laganiere, 2007; Furukawa and Ponce, 2009) to obtain semantic segmentation in buildings. Then, they apply appropriate optimization algorithms to obtain the floorplan. Cabral and Furukawa (2014) cast the optimization as the shortest path problem, and Pintore et al. (2018) obtained the floorplan by solving a nonlinear least squares problem. However, these solutions only handle closed topologies and rely on the quality of the point cloud. In addition, indoor reconstruction provides different schemes using point clouds as input. In view of the fact that the wall is generally perpendicular to the ground, some work (Han et al., 2021; Ochmann et al., 2016, 2019; Cui et al., 2019) first fit vertical planes from the point cloud, followed by projecting them to the ground to generate line segments. The 2D space is then divided into smaller cells by extending line segments. Finally, the floorplan is obtained by minimizing an energy optimization function. Some of them (Ochmann et al., 2016, 2019; Cui et al., 2019) used room segmentation to construct the data item and marked the state of cells, followed by merging 'active' cells as final rooms after optimization. Han et al. (2021), the most relevant work to us, used the indoor facade point cloud to support the data item and obtained the floorplan by marking and selecting the segments with the 'true' label. However, these methods are sensitive to plane detection and lack sufficient robustness to imperfect data. Compared with Han et al. (2021), we deeply integrated 2D plane instances inferred from images into 3D plane detection and optimization, making the method have higher robustness and more accurate reconstruction results for real indoor scenes.

End-to-end networks. With the development of neural networks, end-to-end learning has been widely applied in scene modeling. Considering that panorama provides a wide range of contexts, lots of work (Sun et al., 2021; Pintore et al., 2020a; Zou et al., 2018; Yang et al., 2019; Sun et al., 2019) have used an end-to-end network to predict the layout from panoramic images. Some work (Yang et al., 2019; Pintore et al., 2020a) transformed the panorama to other views from which the floorplan was inferred. Yang et al. (2019) fused features extracted from the original panoramic view and ceiling view, and output the floorplan probability maps, which were then regularized to obtain the layout with Manhattan restriction. Pintore et al. (2020a) used a single branch to predict dense 2D segmented maps from ceiling and floor views, followed by a simple operation to obtain layouts. Instead of combining multiple views, Sun et al. (2019, 2021) represented the input as a 1D vector, where each dimension stored the prediction information relating to the corresponding column in the image. Sun et al. (2019) first used LSTM (Hochreiter and Schmidhuber, 1997) to capture global information and output three 1D vectors representing the scene boundaries. The layout was obtained after postprocessing. Later, Sun et al. (2021) introduced a new horizon-to-dense module to expand each dimension of the vector to all horizontal pixels, and finally obtained the dense panoramic depth map. The abovementioned methods recovered the layout from a single panorama, making it difficult for them to handle large indoor scenes such as shopping malls.

Hybrid schemes. Due to the great potential of deep learning in extracting features and the better interpretability of optimization techniques, many methods employ a hybrid technical framework. Some state-of-the-art work (Liu et al., 2018; Chen et al., 2019; Phalak et al., 2020; Stekovic et al., 2021) first inferred primitive information (e.g., corners, edges, room segments) using a network and then applied optimization strategies to obtain floorplans. Chen et al. (2019) and Stekovic et al. (2021) took the density map projected from the point cloud as input, and obtained the room segments by relying on MaskRCNN (He et al., 2017). The former (Chen et al., 2019) integrated the corner and edge likelihoods from DRN (Yu et al., 2017) into an optimization function with the room segments as constraints, and solved the problem via the sequential roomwise shortest path. The latter (Stekovic et al., 2021) obtained a set of room proposals from room segments and then used the Monte Carlo Tree Search algorithm (Browne et al., 2012; Coulom, 2006) to find the optimal subset and refine their position and shape. Limited by the image resolution, it is difficult for these methods to capture detailed structures. Rather than relying on the point density map, Fang et al. (2021) detected vertical planes from point clouds and projected them onto the X–Y plane to partition the 2D space into smaller cells. They first extracted the scene boundaries and then generated the floorplan via graph-cut (Boykov et al., 2001) using room segments inferred by the network as the unary term. Solarte et al. (2021) embedded the single room layout extracted from the network into a visual SLAM system, and estimated multiroom layout geometries. Our method belongs to this category, and thanks to the combination of 2D semantics and 3D geometry, our approach offers an effective pipeline to obtain floorplans robustly and in detail.

3. Overview

Our algorithm takes the calibrated images and dense mesh where the Z-axis is aligned with the gravity direction as input and outputs the floorplan. The core idea behind our method is to deeply integrate the 2D plane instances and the 3D plane primitives into a unified framework.

The proposed pipeline has three phases: scene segmentation, indoor facade candidate generation, and indoor facade selection (see Fig. 1). First, a semantic segmentation network is used to segment images and we segment indoor facades from point cloud sampling from the dense mesh. The labels of the point cloud are obtained by projecting points onto their visible images and max-voting. Then, we segment plane instances from images via a plane detection network, followed by detecting and regularizing 3D planes from indoor facade point clouds holistically and locally using plane instances. Finally, we design a global optimization function combining plane confidences and geometric quality of the point cloud and obtain the floorplan by minimizing the energy function with integer linear programming with constraints.

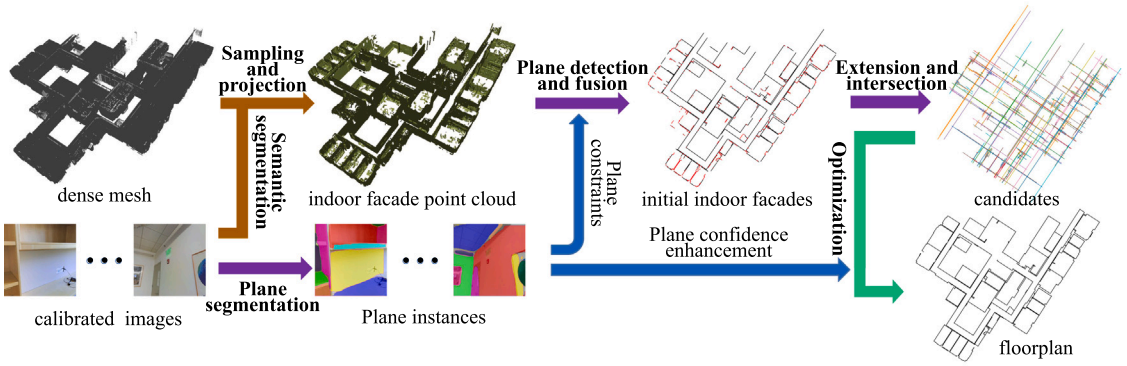

Fig. 1. Overview of FloorUSG. There are three phases: scene segmentation (brown), indoor facade candidate generation (purple) and indoor facade selection (green). The blue arrows indicate that plane instances are fused at different phases. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

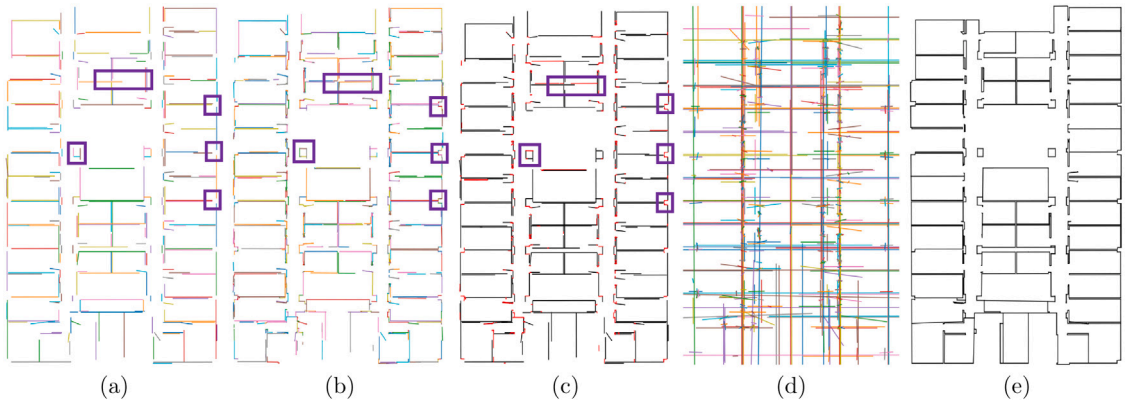

Fig. 2. An example of indoor facade candidate generation and selection on Area 6 of the S3DIS dataset (Armeni et al., 2017). (a) is the segment set detected from the indoor facade point cloud using RANSAC (Schnabel et al., 2007) directly. (b) is the segment set derived from 2D plane instances. (c) is the segment set after fusing (a) and (b). The black represents the general scene structures, which comes from (a). The red is from part of (b) and complements more details, especially the areas circled with purple boxes. (d) is candidates by extension and intersection from (c). (e) is the final floorplan after optimization. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4. Scene segmentation

Indoor scenes often contain clutter that interferes with indoor facade detection and inference. Therefore, we first segment the indoor facade from the input mesh. There are many selectable strategies, such as direct segmentation on 3D data, or segmentation on images first and then projection onto the mesh to fuse labels. Considering that we need to know the correspondence between 3D data and image pixels, here we use the public semantic segmentation network DeepLabv3+ (Chen et al., 2018) to segment images and uniformly sample the input mesh in space as a point cloud. Then, we compute the visibility of points relying on the mesh. In detail, if a line segment s passes through a point and a camera optical center while it does not go through other facets on the mesh, this point is visible in the image which intersects the segment s, and the label of this point is marked as the same as the pixel which is the intersection of image and segment s. The correspondence between this point and the pixel is retained for the next stage. A point may be visible in multiple images, and the final label of the point is determined via max-voting. We take the points labeled as indoor facades as the subsequent input. It should be noted that due to the subsequent fusion and global optimization steps, we do not need very high-quality indoor facade segmentation results, as shown in the experiments.

5. Indoor facade candidate generation

At this stage, we aim to combine the indoor facade point cloud and images to generate as complete candidate line segments as possible. RANSAC (Schnabel et al., 2007) can detect accurate planes from point clouds with noise and outliers. However, this probabilistic approach may miss small planes or planes with sparse supporting points, especially when handling large-scale point clouds. Therefore, we add the plane instances inferred from images to enhance the detection results of traditional RANSAC (Schnabel et al., 2007).

5.1. Indoor facade candidates from point cloud

In view of the outliers and noise in the point cloud, we first use RANSAC (Schnabel et al., 2007) to fit vertical planes from the indoor facade point cloud, followed by projecting them onto the X–Y plane to obtain the line segments. These projected segments contain some unsatisfactory detection and thus, it is necessary to regularize them to produce cleaner results. In principle, two segments that are close enough and have a small angle are more likely to belong to a true segment. Therefore, we merge two line segments s_i and s_j when the following two conditions are satisfied:

$$\theta_{ij} \leq \theta_1, \tag{1}$$

$$dis_{ij} \leq \alpha \cdot \min(d(s_i), d(s_j)), \tag{2}$$

where θ_{ij} and dis_{ij} are the angle and distance between s_i and s_j respectively. $d(s)$ is the average distance between the 2D supporting points of line segments s_i and s_j, and we set $\theta_1 = 10°$ and $\alpha = 5$ in our experiments.

Specifically, the two segments s_i and s_j with the minimum angle are considered first. If they meet the above conditions, we fit a plane from their supporting points using PCA (Wold et al., 1987) and acquire a new segment s_k by projection. The above process is iterated until no segment satisfies the conditions, and finally, we obtain a trim line segment set S_1 (see Fig. 2(a)). The obtained set S_1 contains large indoor facade structures while missing many details.

5.2. Indoor facade candidates from images

The RGB image contains rich semantic information, which has great potential to enhance geometric detection. Lots of work (Yu et al., 2019; Liu et al., 2019; Qian and Furukawa, 2020) recover 3D scenes with predicted 2D plane instances and their 3D parameters from a single image. The task of predicting 3D plane parameters from just a single image is difficult to generalize well on various datasets, however, owing to the advantages of deep learning in image processing, robust 2D plane instance segmentation is feasible.

We use publicly available PlaneRCNN (Liu et al., 2019), which also targets indoor scenes, to segment images. Since the results of plane detection are not completely accurate, we only retain instances with a 'valid' pixel number greater than $\epsilon \cdot \#nums$, and project them onto the indoor facade point cloud to obtain corresponding point sets. Then, we perform local RANSAC (Schnabel et al., 2007) on each point set, followed by projecting fitted planes onto the ground to obtain another line segment set. In our experiments, we set $\epsilon = 5\%$ and $\#nums$ is the number of pixels in an image. Note that the 'valid' here means that the pixels should belong to the indoor facade label. In addition, the point set corresponding to a 2D plane instance may contain some noise as a result of incorrect image inference. Thus, instead of PCA (Wold et al., 1987), which is more susceptible to noise, we adopt local RANSAC (Schnabel et al., 2007) to fit the most likely 3D plane from the point set.

When there are many common viewing regions between images, one 3D plane may be detected in multiple images, which is unknown to the plane segmentation network. As a result, there is considerable redundancy in the above line segment set (e.g. one 3D plane may be represented by multiple segments that are very close together), and here we adopt a fast reduction strategy to clean them. We begin with the X-axis, and produce $180°/\theta_2$ bins at intervals of θ_2. Then we calculate the angle between segments and the X-axis (taking the angle within $180°$) and put them in the corresponding bin. Starting from the first bin, we iteratively merge the segments in the two adjacent bins if the segments satisfy Eqs. (1) and (2). θ_2 is set to the same value as θ_1 in Eq. (1). Finally, we obtain the second line segment set S_2 (see Fig. 2(b)). The obtained set S_2 contains more detailed structures but lacks globality due to the local plane fitting.

5.3. Fusion of different candidate sets

Using RANSAC (Schnabel et al., 2007) to fit planes directly from 3D point clouds makes use of more global information in the scene. It has a strong anti-noise ability, but poor robustness in detecting small structures. In contrast, the method of fitting and reducing 3D planes from 2D plane instances pays more attention to the local scene structures. It has a stronger ability to recover details, while having a lower tolerance to noise. As seen in Fig. 2(b), the obtained segments retain some small structures of scenes. However, long segments tend to be detected as several small ones owing to the local view of images, which is not conducive to the subsequent optimization. In view of these, we implement the following strategy to combine the strengths of both.

First, we merge the image detected planes into the geometric detected planes. For each segment $s_i \in S_2$, if $\exists\ s_j \in S_1$, where s_i and s_j satisfy Eqs. (1) and (2), we delete s_i from S_2 and merge its supporting point set P_i into the set P_j of s_j, followed by marking the plane ID ID_k of point $p_k \in P_i$ as j.

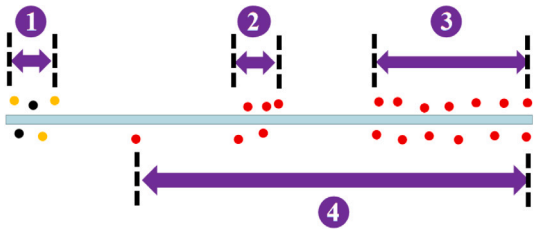

Fig. 3. Explanations of valid lengths in different energy terms in Eq. (3). The points with different colors are 2D supporting points of the candidate (colored with light green). In the process of inferring plane instances from images, the red and yellow points belong to two inferred planes and the black points belong to no plane. As shown in the figure, the valid lengths are different in different energy terms. The lengths marked as 1, 2, and 3 are valid covered lengths in Eq. (6) while others are invalid because the distance between two points is larger than the threshold. The length marked as 4 is the valid confidence length in Eq. (7). Because the number of red points is greater, the red points are picked and the length of the red points is defined as the valid length. By introducing E_3 (Eq. (7)), our method increases the attention to segments with uneven or sparse supporting points. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Next, we collect unconsolidated image detected planes and add them into the segment candidate sets. We traverse the point p in the indoor facade point cloud that meets $\forall P_i,\ s_i \in S_1,\ p \notin P_i$. If this point p is the supporting point of segment s_j, $s_j \in S_2$, we add s_j into the set S_1 and delete s_j from S_2. The plane ID ID_k of point $p_k \in P_j$ is marked as j.

Finally, we obtain the line segment set $S = S_1$ containing more details and global information. As shown in Fig. 2(c), the black segments are from the initial S_1, which detect the general scene structures. The red segments are from the initial S_2, which find more details, especially in the areas highlighted by purple boxes. We extend segment $s_i \in S$ to a certain length, and use the 2D boundary box of the scene to crop them. The complete indoor facade candidate set F is generated by calculating the intersection of each two line segments (see Fig. 2(d)).

The fusion of plane semantics inferred from images and direct geometric detection helps to obtain a complete indoor facade candidate set more robustly than the traditional plane detection method and reduces the adverse effects of uneven or sparse point cloud density on the subsequent optimization.

6. Indoor facade selection

After the above stage, we obtain the candidate line segment set $F = \{f_i\}$ and the corresponding supporting point set $P = \{P_i\}$. Now, we aim to select the optimal subset from the set F to form the final floorplan.

We define an energy optimization function and the floorplan is obtained by minimizing this function. Han et al. (2021) adopted a similar strategy to solve the floorplan reconstruction, but they only made use of the quality of the point cloud. As a result, their approach had a weak ability to distinguish the false plane with noise and the true plane with sparse supporting points. In contrast, the proposed optimization function in our paper takes into account the high-level semantics of images and increases the attention of small planes and planes with sparse supporting points.

6.1. Objective function

We take both the quality of the point cloud and the plane instances inferred from images into consideration for the sake of better data balance. In detail, we introduce a binary variable $x_i \in \{0, 1\}$ for each candidate $f_i \in F$, and design an energy function E consisting of four items: point fidelity term E_1, point coverage term E_2, plane confidence

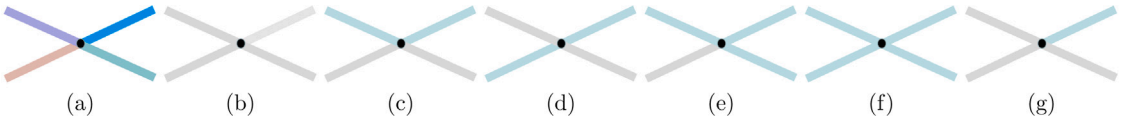

Fig. 4. Examples of the different selections of segments connected by one intersection. (a) displays four segments sharing one intersection v_i. In others, the light green represents that the segment is selected, while the gray means the opposite. For the sake of model closure, (g) is forbidden and the others are allowed. In addition, (c), (e) and (f) introduce a corner with $Corner(v_i) = 1$.

term E_3, and model complexity term E_4 with $\{x_i\}$ as the independent variable:

$$E = \sum_{k=1}^{4} \lambda_k \cdot E_k, \tag{3}$$

where λ_k is the balance factor, and we set λ_1 to 0.4 with the rest to 0.2 in our experiments.

Our goal is to minimize the function E by determining the value of $\{x_i\}$. The final floorplan is obtained after selecting candidate f_i with the value $x_i = 1$.

Point fidelity term E_1. This is the most basic term of the function, reflecting the supporting strength and the fitting accuracy of the supporting point set P_i to the candidate f_i. The term E_1 is defined as:

$$E_1 = 1 - \frac{1}{N_p} \sum_{i=1}^{N_f} \Big(\sum_{p_j \in P_i} 1 - \frac{\min(dis(f_i, p_j), \rho)}{\rho} \Big) \cdot x_i, \tag{4}$$

where N_p is the total number of points in P, N_f is the number of candidates, $dis(f_i, p_j)$ is the distance between the 2D projection of point p_j and segment f_i, and ρ is a distance threshold providing a uniform constraint for all candidate segments.

Specifically, for each segment f_i, the point p_j is considered only when $dis(f_i, p_j) < \rho$. The smaller the $dis(f_i, p_j)$ is, the more accurate the point-to-segment fitting, and the more supporting points that satisfy the distance condition, the stronger the point-to-segment support. When points fully support and fit all line segments, the point fidelity can be maximized, corresponding to the minimum $E_1 = 0$. In our experiments, we set $\rho = 3 \cdot dis(f)$ with:

$$dis(f) = \frac{1}{N_p} \sum_{f_i \in F} \sum_{p_j \in P_i} dis(f_i, p_j). \tag{5}$$

Point coverage term E_2. Due to the occlusion and the weak textures in real scenes, the obtained point cloud inevitably contains some missing areas, which should also be considered during reconstruction. This term is designed to balance the noise and missing in the point cloud, which is defined as:

$$E_2 = \frac{1}{N_f} \cdot \sum_{i=1}^{N_f} \Big(1 - \frac{len_{cov}(f_i)}{len(f_i)}\Big) \cdot x_i, \tag{6}$$

where $len(f_i)$ is the length of segment f_i, and $len_{cov}(f_i)$ is its covered length.

The 2D supporting points are projected to the corresponding segment f_i to obtain the projection set PP_i, and the distribution of PP_i reflects the extent to which the segment f_i is covered. We calculate the distance between adjacent points in PP_i and mark the distance as valid if it is less than $\mu \cdot den$ (den is the density of supporting points). The $len_{cov}(f_i)$ is obtained by adding up all the valid distances. When points cover all segments, the point coverage can be maximized, corresponding to the minimum $E_2 = 0$. In our experiments, we set $\mu = 5$.

Plane confidence term E_3. The above two energy terms mainly evaluate the quality of the point cloud. For some noise and regions with sparse supporting points in the scene, it is difficult to effectively distinguish them only by considering the geometric characteristics of the point cloud.

Thus, we introduce this term, which measures how much confidence a candidate belongs to a plane instance inferred from images, to boost the robustness of the method. It is defined as:

$$E_3 = 1 - \frac{1}{N_f} \cdot \sum_{i=1}^{N_f} \Big(\frac{N^i_{conf}}{N^i_p} \cdot \frac{len_{conf}(f_i)}{len(f_i)}\Big) \cdot x_i, \tag{7}$$

where N^i_p is the number of supporting points of segment f_i.

Theoretically, a candidate corresponds to one or zero planes, however, due to the fact that the instance segmentation network is not completely accurate, a candidate may contain supporting points belonging to more than one plane instance. Thus, in the supporting point set of candidate f_i, we group the points with the same plane ID into a cluster and pick one cluster with the largest number of points. N^i_{conf} is defined as the size of this cluster, and $len_{conf}(f_i)$ is defined as the maximum distance of the projection of two points in this cluster on f_i. In this term, we take into account the number and distribution of points, and when a segment f_i is fully covered by just one inferred plane, the plane confidence can be maximized, corresponding to the minimum $E_3 = 0$. (see Fig. 3 for further explanation.)

Model complexity term E_4. This term is considered to balance the model fidelity and complexity. Here, we measure the complexity of the floorplan with the number of corners, and the more corners there are, the more complex the model.

In general, an intersection v_i is connected to four line segments (except the boundary intersection). In these four segments, a corner is introduced when two noncollinear segments are added to the floorplan (see Fig. 4(b)(c)(e)), and we mark $Corner(v_i) = 1$. Otherwise, $Corner(v_i)$ is set to 0. We calculate the number of intersections (as N_v) and define this term as:

$$E_4 = \frac{1}{N_v} \cdot \sum_{j=1}^{N_v} Corner(v_j). \tag{8}$$

Constraint. The structural characteristics of indoor scenes may be quite different. For complex large scenes, such as office buildings and shopping malls, one thick indoor facade may connect multiple rooms. To guarantee the closure of the scenes, we restrict the selection of segments connected by one intersection $v_i \in V$ to not one (see an example in Fig. 4(g)):

$$\forall v_i \in V, \sum_{f_j \in neig(v_i)} x_j = 0 \ or \ 2 \ or \ 3 \ or \ 4. \tag{9}$$

For small-scale home scenes composed of some independent rooms, the 2-manifold of the house is more conducive to the generation of topologically consistent structures. Therefore, when dealing with such scenarios, we constrain the 2-manifold of models as follows:

$$\forall v_i \in V, \sum_{f_j \in neig(v_i)} x_j = 0 \ or \ 2, \tag{10}$$

where $neig(v_i)$ stores the segments connected by the intersection v_i.

6.2. Optimization

The floorplan reconstruction is formulated as the following optimization:

$$\min_X E \quad s.t. \begin{cases} \text{Eq. (9)} \quad or \quad \text{Eq. (10)} \\ x_i \in \{0, 1\} \qquad\quad 0 < i \le N_f \end{cases} \tag{11}$$

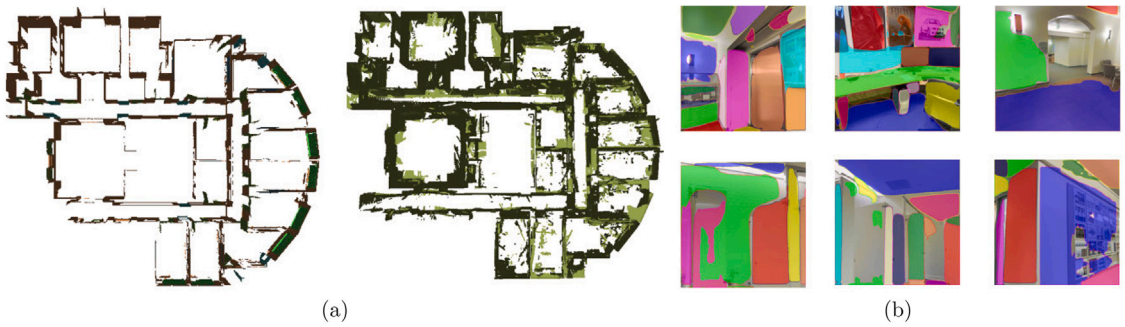

Fig. 5. Scene segmentation and plane detection on Area 3 of S3DIS (Armeni et al., 2017). (a) displays the indoor facade ground truth (left) and our segmentation result (right). (b) displays some plane detection results using PlaneRCNN (Liu et al., 2019) without fine-tuning. As seen, there are still many visible errors and noises in scene segmentation and plane detection. However, due to the multilevel plane discovery and fusion and the global optimization, our algorithm is robust for imperfect inferences from the network.

The Eq. (11) is an integer linear programming problem, and we solve it using SCIP (Bestuzheva et al., 2021). The constraint is determined according to the characteristics of the scene. Finally, we put segments with $x_i = 1$ together to obtain the final floorplan (see Fig. 2(e)).

7. Experiments

To comprehensively evaluate the effectiveness of FloorUSG, we evaluated it on two datasets with different scales and scene structure characteristics, and compared it with other state-of-the-art methods. The algorithm was implemented in C++ with the CGAL Library (The CGAL Project, 2022) and the SCIP solver (Bestuzheva et al., 2021). All the experiments were performed on a PC with a 4-core Intel Xeon CPU (3.7 GHz).

7.1. Dataset

FloorUSG was evaluated on two datasets. The first is the S3DIS dataset (Armeni et al., 2017), which is a large 2D–3D-semantics dataset with a total of 6 large-scale indoor areas (Area 1-Area 6) and 13 object classes, and includes offices, lobbies, rooms, exhibition areas, open spaces, and so on. In all areas, the floor space ranges from 450 m^2 (Area 3) to 1700 m^2 (Area 5) and the number of disjoint spaces ranges from 24 (Area 3) to 55 (Area 5). The RGB images and corresponding dense meshes are provided in the dataset and we used them as the input of our algorithm. In addition, we sampled the dense mesh uniformly in space as a point cloud and took the points with the ground truth *wall, door, window, column, board* labels as our indoor facade ground truth.

The second dataset (we named it HOUSE) includes 100 panoramic RGB-D scans of small indoor scenes and 2D floorplan ground truth provided by FloorSP (Chen et al., 2019). Each scene contains multiple enclosed rooms with an average number of approximately 7, and the scene area ranges from approximately 40 m^2 to 300 m^2. Compared with HOUSE, the scenes of S3DIS (Armeni et al., 2017) are larger with more complex structures. On the HOUSE dataset, we used Poisson surface reconstruction (Kazhdan et al., 2006) to generate dense meshes from point clouds derived from RGB-D scans, and took RGB images and dense meshes as our input.

7.2. Implementation details

7.2.1. Scene segmentation

We used pretrained DeepLabv3+ (Chen et al., 2018) trained on ImageNet (Deng et al., 2009) to segment RGB images on two datasets. To make the network more suitable for each area and scene labels, we randomly selected 50 images on each area of S3DIS and 10 images on each area of HOUSE to fine-tune the network. Then, we sampled the mesh uniformly in space as a point cloud with a sampling size $\delta = 0.02$ m, and projected points onto their visible images. If a point was visible in multiple images, its label was determined by max-voting. Because we only focused on indoor facades, 3D points with labels *wall, door, window, column, board* were retained. The segmentation errors between these five labels had little effect on our method and the obtained indoor facade point cloud was sufficient for our subsequent input owing to the robustness of our method to imperfect data (see Fig. 5(a) as an example).

7.2.2. Plane segmentation

We used PlaneRCNN (Liu et al., 2019) with the default hyperparameter to detect plane instances of RGB images on two datasets. The provided model in PlaneRCNN has been pretrained on Scannet (Dai et al., 2017), which is also an indoor dataset that mainly contains single rooms. The scenes in all three datasets are similar, and the detection results on S3DIS and HOUSE without fine-tuning were acceptable. Furthermore, due to the content redundancy between images and the effective fusion in our pipeline, a few segmentation errors were tolerated (see Fig. 5(b) as an example).

7.2.3. Parameter selection

Our method embeds the high-level semantic information inferred through deep learning into the reconstruction process based on geometric optimization. The key parameters in the method include line extending parameter, plane fitting parameters and the energy item weights in the optimization function. Extending parameter is important to the selection of the candidate segments. Large extension will create more candidate line segments, at the same time greatly increasing the complexity of the calculation and increasing the optimization time because too many segments increase the difficulty of optimization, while a small extension will miss planes and cannot obtain the complete results. We take the first area in the S3DIS dataset as an example and record the floorplan of different extending parameters. Fig. 6 shows the different results by gradually increasing the extending parameter. As shown in the figure, a small extending parameter produces a floorplan with many missing segments (see green and purple boxes in Fig. 6(a)), and a large extending parameter obtains an approximate effect with the parameter we selected, while a moderate extending parameter is the best. In addition, the optimization time increases with increasing extending parameter, when the parameters are 0, 0.15 and 0.3, the optimization time are 484 s, 646 s and 1521 s respectively. For the plane detection, the major parameter affecting the quality and quantity of the detected plane is *min_points* in RANSAC which represents the minimum number of supporting points in a plane. The higher the parameter is, the larger and fewer planes are detected, but some small planes may be missed. The lower the parameter is, the smaller and more planes are detected, but they may contain some noise. When dealing with real indoor scenes with a point cloud density of 0.02 m, it is generally appropriate to set this value to 500 to 2000. Considering that we need to integrate large planes directly fitted from the point cloud and small planes inferred from the image as complementary, we

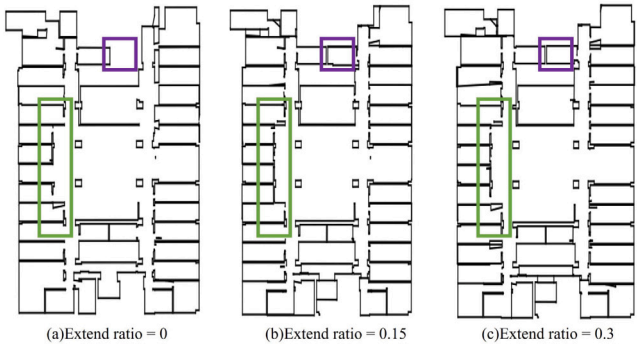

Fig. 6. Effect of different extending parameters. (a) is the floorplan when the segments are not extended(0), (b) is the floorplan when the extending ratio is moderate(0.15), and (c) is the floorplan when the extending ratio is large(0.3). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

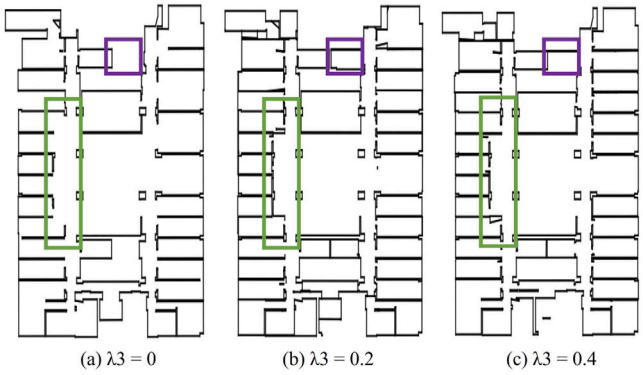

Fig. 7. Effect of different confidence of the image detected plane weight λ_3 in Eq. (3) on S3DIS Area1. (a) is the floorplan when image detected planes are not added. (b) is the floorplan with the weight we choose(0.2), and (c) is the floorplan with a large confidence weight(0.4). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

set *min_points* to 1000 when detecting large planes, and set it to 500 when detecting small planes. In addition, other parameters that affect the quality of the floorplan are the four energy term weights in Eq. (3). These weights can be adjusted slightly to fit point clouds of different qualities. In the experiments, we all use the default set, that is, $\lambda_1 = 0.4$, $\lambda_2 = 0.2$, $\lambda_3 = 0.2$ and $\lambda_4 = 0.2$. In Eq. (3), the terms E_1, E_2 and E_4 are essential for generating a reasonable result. Compared with the method proposed in VecIM (Han et al., 2021), the third term E_3 is added to control the confidence of the image detected planes. Fig. 7 shows the different results by gradually increasing the confidence weight λ_3 with $\lambda_1 = 1 - \lambda_2 - \lambda_3 - \lambda_4$, $\lambda_2 = 0.2$, $\lambda_4 = 0.2$. When λ_3 is small, the small plane missing in the floorplan is more serious (see green and purple boxes in Fig. 7(a)). When λ_3 is large, parts of the floorplan detected by geometric plane detection but not detected in the picture plane detection are missing (see purple boxes in Fig. 7(c)).

7.3. Evaluations on the S3DIS dataset

The S3DIS dataset includes office areas, corridors, open spaces, etc., with large scale and complex structures. These scenes have difficulty guaranteeing the 2-manifolds, and thus, we used Eq. (9) as the constraint and ensured the closure of the floorplan.

7.3.1. Qualitative evaluations

In this section, we qualitatively compared the results of FloorUSG with some state-of-the-art approaches (Chen et al., 2019; Han et al., 2021; Liu et al., 2018) on the first four areas of S3DIS (see Fig. 8). Since Han et al. (2021) (hereinafter referred to as VecIM) needed to segment the indoor facade from the point cloud as preprocessing, we took our segmented points as their input. FloorNet (Liu et al., 2018) and FloorSP (Chen et al., 2019) took the point cloud of the whole scene as input, and we uniformly sampled the dense mesh and input the sampling point cloud into their pipeline.

FloorNet (Liu et al., 2018) and FloorSP (Chen et al., 2019) first inferred primitive information via a neural network and obtained floorplans by optimization. As seen in Fig. 8, their results were not satisfactory. This was largely due to the poor primitive inference on the point density map. FloorNet only handled scenes with the Manhattan assumption, and the two methods applied a low-resolution map (256*256), which severely limited the complexity of scenes and the recovery of detailed structures. Instead of relying on delicate corner/edge/room detection on the planar graph, we effectively utilized the 2D plane instances to enhance the geometric detection and optimization, making us reconstruct more accurate results on large-scale scenes.

VecIM (Han et al., 2021) adopted a pure geometric optimization method, which mainly relied on RANSAC (Schnabel et al., 2007) to detect primitives and obtained the floorplan via a global optimization. Due to the detection instability of RANSAC and missing points in the point cloud, it was not easy for this method to reconstruct real scenes with complex structures robustly and completely. In contrast, we took into account the 2D semantics and deeply integrated it with the 3D geometry of the point cloud, making our method achieve better reconstruction results. On the one hand, we used 2D plane instances to effectively supplement the missing detection of RANSAC (see red segments in the second row of Fig. 8). The fusion of 2D and 3D primitives gave our method a stronger ability to capture scene details, even in large-scale scenes with complex structures. On the other hand, we increased the selection probability of weak but real structures by combining both the plane confidence of points and the quality of point cloud during optimization (see green boxes in Figs. 8,9). This allowed our method to offer high robustness for imperfect data.

7.3.2. Quantitative evaluations

Considering that there is no 2D floorplan ground truth and that both the ceiling and floor are horizontal on S3DIS, we restored 3D models by lifting the floorplan of ours and VecIM to the average height of the ceiling and floor on the last two areas of S3DIS. The 3D models were compared to the indoor facade ground truth. Here, we also compared the results of Polyfit (Nan and Wonka, 2017), which is a general method for reconstructing 3D holistic models. Since this method is suitable for closed scenes and the clutter in the scene disturbs the plane detection, we uniformly sampled the mesh in space with the ground truth *ceiling, floor, wall, door, window, column, board* labels and took the sampling point cloud as the input of Polyfit.

For quantitative comparison, we calculated the Hausdorff distance from the indoor facade ground truth to the 3D model as the reconstruction error. As seen in Fig. 10, our models are closest to the 3D ground truth with the lowest mean error and root mean square. Due to the two-manifold constraint forced by Polyfit, at least one room failed to be recovered when a thin wall was shared by two rooms. In addition, similar to VecIM, Polyfit is also a method that only considers the geometry of data, making it difficult to robustly handle data of different qualities. In contrast, our approach fused high-level semantics at different phases to increase the robustness of the algorithm and the reconstruction quality of models.

7.4. Evaluations on the HOUSE dataset

7.4.1. Qualitative evaluations

The public House dataset contains 100 houses. Among all the data, we selected scenes with different complexity and characteristics for evaluation and made quantitative and qualitative comparisons to prove

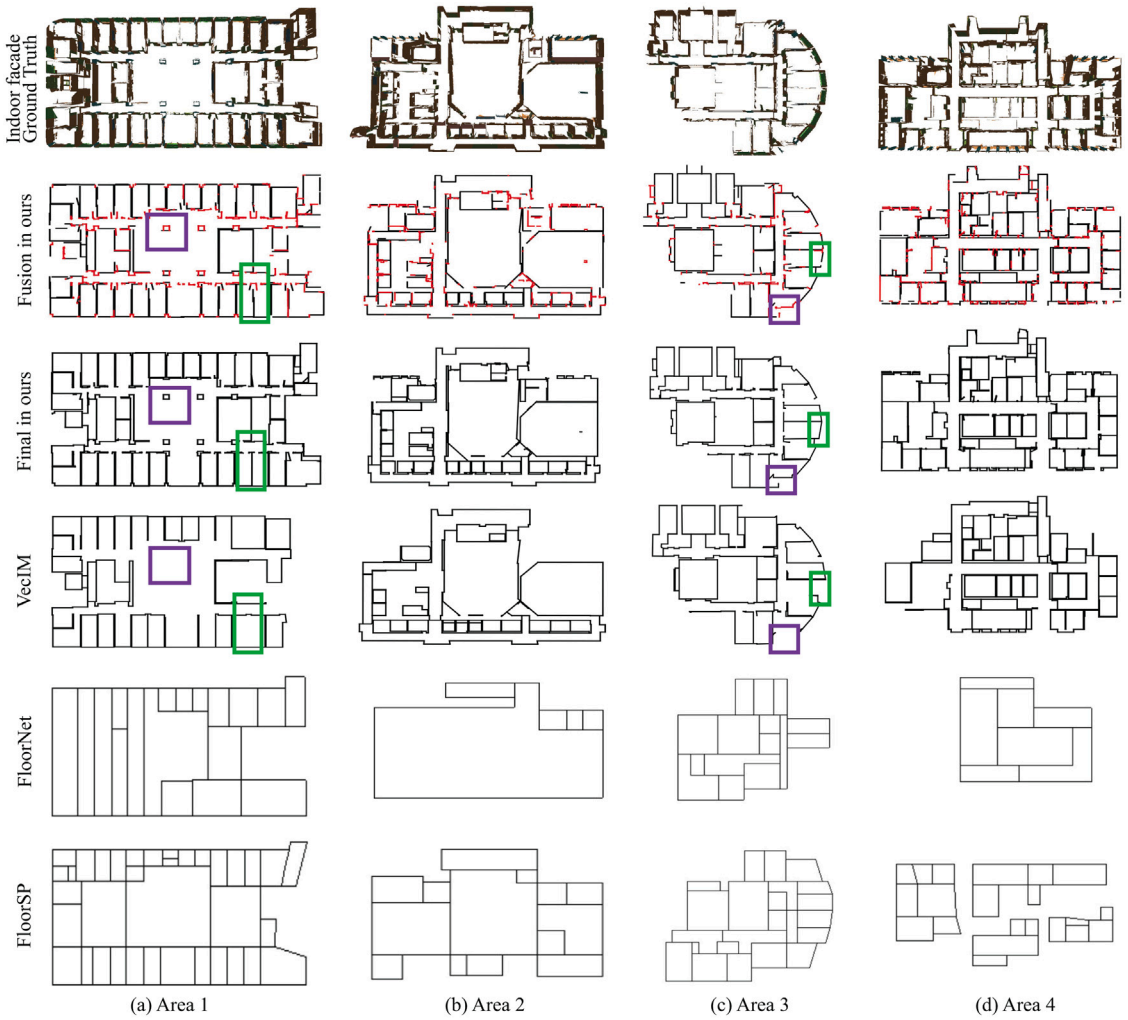

Fig. 8. Qualitative comparisons on S3DIS. The first row shows the indoor facade ground truth. The second row shows the fusion of candidates in our method. The black is from direct RANSAC, and the red is from image inference which is a great supplement. The third row is our final floorplan. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

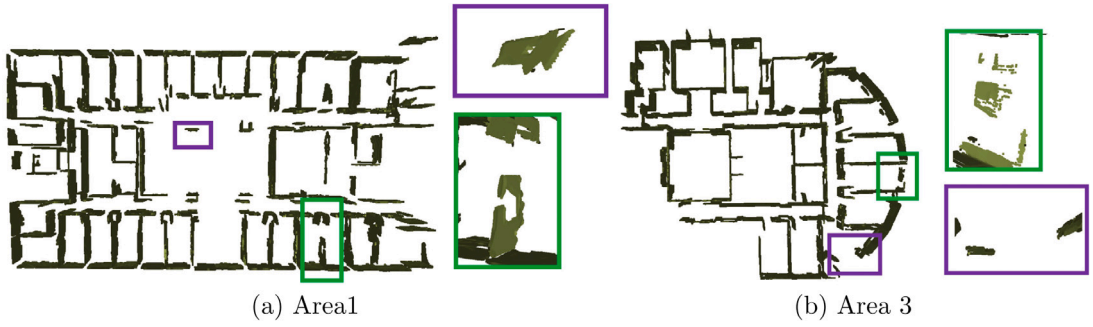

Fig. 9. RANSAC results of VecIM (Han et al., 2021) on S3DIS. The purple emphasizes the missing plane detection, and the green emphasizes the plane detection with sparse supporting points. In these conditions, VecIM lost the detailed structures while our method recovered these areas more completely and accurately (see the third and fourth rows of Fig. 8). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the robustness and reconstruction accuracy of the method in such scenes. Due to the observation that the HOUSE dataset mainly contains several small closed rooms with relatively simple structures, we used Eq. (10) to restrict the 2-manifold of the scenes, that is, to ensure that the walls of each room were not shared with other rooms.

Considering that FloorNet (Liu et al., 2018) only dealt with Manhattan scenes and the semantic segmentation results on the HOUSE dataset were poor which severely degraded the quality of the generated floorplan, we only compared the proposed method with VecIM (Han et al., 2021) and FloorSP (Chen et al., 2019) on this dataset. As in the S3DIS dataset, we used the segmented facade points as the input of VecIM and the point cloud uniformly sampled from the dense mesh as the input of FloorSP.

Fig. 11 shows the comparison of the reconstruction results on four scenes in the HOUSE dataset. As seen, FloorSP did not consider the wall thickness and some structural details. The ground truth labeled by

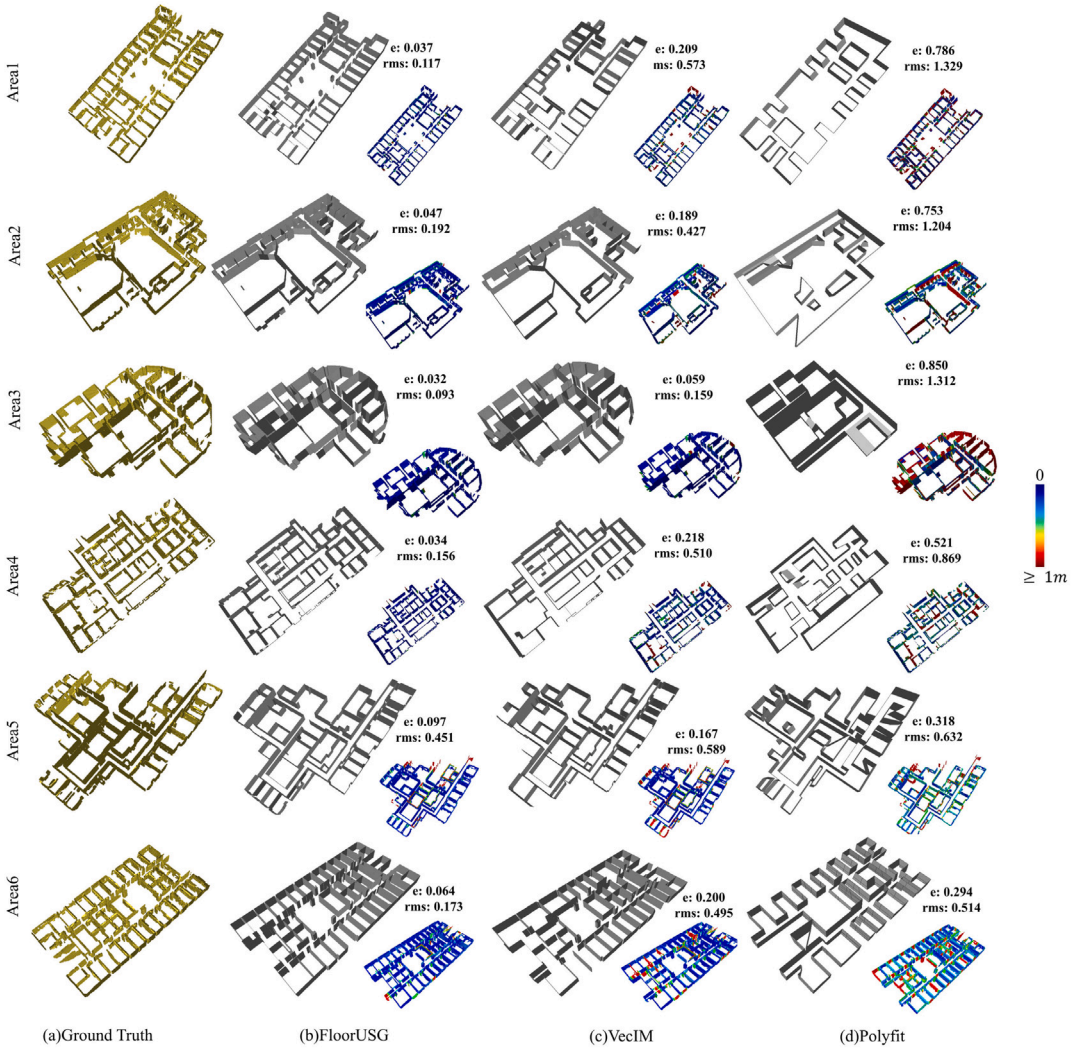

Fig. 10. Quantitative comparison on S3DIS. *e* is the mean of the Hausdorff distance from the indoor facade ground truth to the 3D model, and *rms* is the root mean square. For the same comparison, we removed the ceiling and floor of the model in Polyfit. Compared with others, our method recovered results most accurately.

it paid more attention to restoring the concise outline of the room, and some areas were inconsistent with the scene point cloud. The floorplan generated by this method was close to the ground truth, but the reconstruction results on some non-Manhattan structures were poor, as shown in the last two scenes in Fig. 11. In contrast, our method and VecIM focused on the restoration of more complete and finer grained facade structures, and the results were closer to the real scene and more suitable for some applications that require high reconstruction quality, such as Building Information Modeling (BIM). Due to the addition of indoor facades inferred from RGB images, our method reconstructed more structural details in the scene, such as columns, than VecIM.

Although our method needs to infer plane instances from images, it is a relatively simple task and the inference results of the existing plane detection networks in indoor scenes are not bad. On the other hand, both the plane fusion and the global geometric optimization with different energy terms to balance in our method enhance the robustness and the quality of semantic inference. Therefore, different from many deep-learning methods that heavily rely on training data to guarantee reliable inference, our approach can generate floorplans with good quality including detailed structures whether in large-scale scenes with complex structures or small home scenes.

7.4.2. Quantitative evaluations

The ground truth in the HOUSE dataset pays more attention to restoring the geometric contour and connection relationship of the rooms, while our method pays more attention to restoring the geometric details of the room. Therefore, we do not use the ground truth value of the HOUSE dataset, but restore the floorplan to a three-dimensional structure. Then we compare it with the original point cloud, and calculate its Hausdorff distance to judge the reconstruction accuracy.

For the deep learning method, FloorSP uses a small resolution map (256*256), which results in low resolution reconstruction results and loss of structural details, and its accuracy is lower than the actual physical scale corresponding to a pixel. Therefore, we only compare with VecIM and Polyfit. Similar to the S3DIS dataset, we calculated the Hausdorff distance from the facade point cloud ground truth to the 3D model as the reconstruction error. In the HOUSE dataset, shown in Fig. 12, FloorUSG is also the closest to the ground truth, while the reconstruction results of VecIM and Polyfit are both missing to some degree.

7.5. Running time

We recorded the time spent by our method in each part of the whole process, including the time spent in plane extraction on images, plane extraction directly with RANSAC in point clouds and final optimization in Table 1. As we can see, the image plane detection takes the most time in our method while it takes less time in other steps. However, it

Fig. 11. Comparison of reconstruction results on four scenes in the HOUSE dataset. The first row displays the scene point cloud derived from panoramic RGB-D scans. The second row displays the 2D floorplan ground truth labeled by FloorSP. The other rows are the results of different methods.

is worth noting that the introduction of image plane detection complements small planes, so that geometric plane detection is more targeted at large planes, thus spending less time. In addition, we also tested the time spent by different methods on the S3DIS and HOUSE datasets. Among them, FloorNet makes predictions directly on the images, but it needs to meet the Manhattan hypothesis, and cannot obtain results on some scenes. Therefore, we mainly compare with VecIM, Polyfit and FloorSP. The results are also shown in Table 1. Polyfit is optimized in 3D space, so its time is closely related to the number of fitted planes. We make a compromise between precision and time efficiency. As shown in Table 1, our method can achieve time efficiency similar to VecIM, but is slower than FloorSP.

7.6. Limitations

Although the plane detection in the image is used to supplement the plane detection in geometry, so that the sparse and uneven point cloud information can be effectively used, and the effect has also been verified by experiments, the method also has some limitations. First, the image detected plane is related to the projection of 2D images to 3D meshes, which depends on the quality of the mesh generated by the model. If the initial mesh does not exist somewhere, even if it is detected at the image level, it cannot be reconstructed. Therefore, the sparse part of the point clouds can be supplemented and strengthened, but missing planes with no points cannot be reconstructed, as shown in Fig. 13(a). Second, the purpose of image plane detection is to compensate for the lack of geometric detection at the sparse point clouds. Therefore, when the image fails to cover the sparse point cloud or detect the plane successfully, the missing plane cannot be detected, as shown in Fig. 13(b).

8. Conclusion

We propose an automatic algorithm to reconstruct floorplans from mesh and RGB images. Different from the pure geometric optimization (Han et al., 2021; Nan and Wonka, 2017) and the two-stage approaches (Chen et al., 2019; Liu et al., 2018) relying on the low-resolution point density map, our method embeds 2D plane instances inferred from images into an unambiguously interpretable optimization problem, and has the ability to recover high-quality floorplans in both large-scale complex indoor scenes and small-scale home scenes robustly and accurately, as shown in the experiments.

In future work, we would like to extend the method to recover floorplans with more semantic details, such as doors and windows.

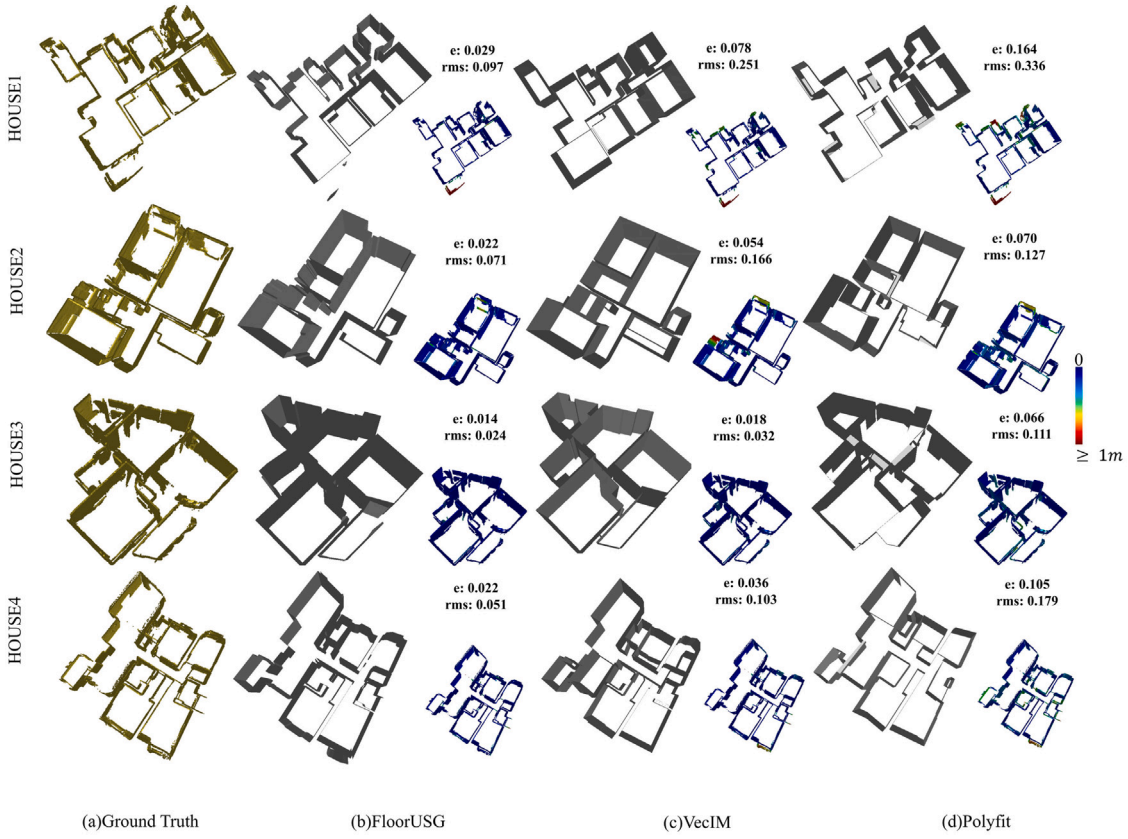

Fig. 12. Quantitative comparison on HOUSE. *e* is the mean of the Hausdorff distance from the indoor facade ground truth to the 3D model, and *rms* is the root mean square. As shown, our method performed better than any other methods.

Table 1
The computational time of each step in our method, as well as the computational time of the compared methods on S3DIS dataset and HOUSE dataset. In the table, $T1$ is the time for image plane detection, $T2$ is the time for geometric plane detection, $T3$ is the time for merging segments, $T4$ is the time for optimization, and TOT is the total running time of our method.

Dataset	$T1$ (s)	$T2$ (s)	$T3$ (s)	$T4$ (s)	TOT (s)	VecIM (s)	Polyfit (s)	FloorSP (s)
S3DIS Area1	485	335	116	646	1582	2057	2340	1421
S3DIS Area2	910	275	66	108	1359	2284	2358	1493
S3DIS Area3	1286	234	29	221	1752	1259	1642	1869
S3DIS Area4	492	159	107	474	1232	2575	2372	2190
S3DIS Area5	722	336	100	328	1486	2604	2890	649
S3DIS Area6	431	294	71	488	1284	2019	2365	2750
HOUSE1	79	107	18	22	226	380	456	810
HOUSE2	104	205	34	31	374	566	648	469
HOUSE3	64	72	7	9	152	286	432	259
HOUSE4	58	103	13	11	185	281	230	1100

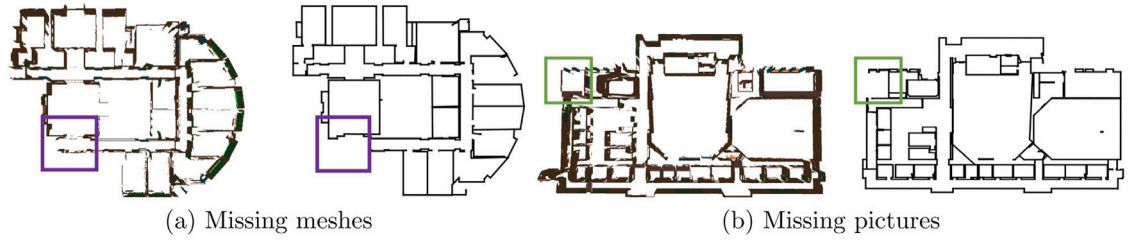

Fig. 13. Limitations: When the mesh of the scene is missing, or there is mesh but the corresponding image is missing, the reconstruction method will be invalid.

In addition, we plan to integrate scene segmentation with plane segmentation into one network to infer images more deeply. Furthermore, although there are many network algorithms to restore the true depth from images, these algorithms do not have good practicability and generalization. The reconstruction of missing areas still needs further exploration, which is also the focus of our future work.

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

Armeni, I., Sax, A., Zamir, A.R., Savarese, S., 2017. Joint 2D-3D-semantic data for indoor S2cene understanding. ArXiv E-Prints arXiv:1702.01105.

Bestuzheva, K., Besançon, M., Chen, W.-K., Chmiela, A., Donkiewicz, T., van Doornmalen, J., Eifler, L., Gaul, O., Gamrath, G., Gleixner, A., Gottwald, L., Graczyk, C., Halbig, K., Hoen, A., Hojny, C., van der Hulst, R., Koch, T., Lübbecke, M., Maher, S.J., Matter, F., Mühmer, E., Müller, B., Pfetsch, M.E., Rehfeldt, D., Schlein, S., Schlösser, F., Serrano, F., Shinano, Y., Sofranac, B., Turner, M., Vigerske, S., Wegscheider, F., Wellner, P., Weninger, D., Witzig, J., 2021. The SCIP Optimization Suite 8.0. ZIB-Report 21–41, Zuse Institute Berlin, URL: http://nbn-resolving.de/urn:nbn:de:0297-zib-85309.

Boniardi, F., Caselitz, T., Kümmerle, R., Burgard, W., 2017. Robust LiDAR-based localization in architectural floor plans. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, pp. 3318–3324.

Boykov, Y., Veksler, O., Zabih, R., 2001. Fast approximate energy minimization via graph cuts. IEEE Trans. Pattern Anal. Mach. Intell. 23 (11), 1222–1239.

Browne, C.B., Powley, E., Whitehouse, D., Lucas, S.M., Cowling, P.I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., Colton, S., 2012. A survey of monte carlo tree search methods. IEEE Trans. Comput. Intell. AI Games 4 (1), 1–43.

Cabral, R., Furukawa, Y., 2014. Piecewise planar and compact floorplan reconstruction from images. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 628–635.

Chen, J., Liu, C., Wu, J., Furukawa, Y., 2019. Floor-sp: Inverse cad for floorplans by sequential room-wise shortest path. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2661–2670.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H., 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In: ECCV.

Coulom, R., 2006. Efficient selectivity and backup operators in Monte-Carlo tree search. In: International Conference on Computers and Games. Springer, pp. 72–83.

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 J. Sel. Top. Appl. Earth Obs. Remote Sens. 12 (8), 3117–3130.

Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M., 2017. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In: Proc. Computer Vision and Pattern Recognition (CVPR). IEEE.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. Ieee, pp. 248–255.

Fang, H., Lafarge, F., Pan, C., Huang, H., 2021. Floorplan generation from 3D point clouds: A space partitioning approach. ISPRS J. Photogramm. Remote Sens. 175, 44–55.

Furukawa, Y., Ponce, J., 2009. Accurate, dense, and robust multiview stereopsis. IEEE Trans. Pattern Anal. Mach. Intell. 32 (8), 1362–1376.

Han, J., Rong, M., Jiang, H., Liu, H., Shen, S., 2021. Vectorized indoor surface reconstruction from 3D point cloud with multistep 2D optimization. ISPRS J. Photogramm. Remote Sens. 177, 57–74.

He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2961–2969.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780.

Kangni, F., Laganiere, R., 2007. Orientation and pose recovery from spherical panoramas. In: 2007 IEEE 11th International Conference on Computer Vision. IEEE, pp. 1–8.

Kazhdan, M., Bolitho, M., Hoppe, H., 2006. Poisson surface reconstruction. In: Proceedings of the Fourth Eurographics Symposium on Geometry Processing. 7.

Liu, C., Kim, K., Gu, J., Furukawa, Y., Kautz, J., 2019. Planercnn: 3d plane detection and reconstruction from a single image. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4450–4459.

Liu, C., Schwing, A.G., Kundu, K., Urtasun, R., Fidler, S., 2015. Rent3d: Floor-plan priors for monocular layout estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3413–3421.

Liu, C., Wu, J., Furukawa, Y., 2018. Floornet: A unified framework for floorplan reconstruction from 3d scans. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 201–217.

Nan, L., Wonka, P., 2017. Polyfit: Polygonal surface reconstruction from point clouds. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2353–2361.

Ochmann, S., Vock, R., Klein, R., 2019. Automatic reconstruction of fully volumetric 3D building models from oriented point clouds. ISPRS J. Photogramm. Remote Sens. 151, 251–262.

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

Phalak, A., Badrinarayanan, V., Rabinovich, A., 2020. Scan2plan: efficient floorplan generation from 3d scans of indoor scenes. arXiv preprint arXiv:2003.07356.

Pintore, G., Agus, M., Gobbetti, E., 2020a. AtlantaNet: Inferring the 3D indoor layout from a single 360° image beyond the manhattan world assumption. In: European Conference on Computer Vision. Springer, pp. 432–448.

Pintore, G., Ganovelli, F., Pintus, R., Scopigno, R., Gobbetti, E., 2018. 3D floor plan recovery from overlapping spherical images. Comput. Vis. Media 4 (4), 367–383.

Pintore, G., Mura, C., Ganovelli, F., Fuentes-Perez, L., Pajarola, R., Gobbetti, E., 2020b. State-of-the-art in automatic 3D reconstruction of structured indoor environments. In: Computer Graphics Forum, Vol. 39. Wiley Online Library, pp. 667–699.

Qian, Y., Furukawa, Y., 2020. Learning pairwise inter-plane relations for piecewise planar reconstruction. In: European Conference on Computer Vision. Springer, pp. 330–345.

Schnabel, R., Wahl, R., Klein, R., 2007. Efficient RANSAC for point-cloud shape detection. In: Computer Graphics Forum, Vol. 26. Wiley Online Library, pp. 214–226.

Solarte, B., Liu, Y.-C., Wu, C.-H., Tsai, Y.-H., Sun, M., 2021. 360-DFPE: Leveraging monocular 360-layouts for direct floor plan estimation. arXiv preprint arXiv:2112.06180.

Stekovic, S., Rad, M., Fraundorfer, F., Lepetit, V., 2021. Montefloor: Extending mcts for reconstructing accurate large-scale floor plans. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16034–16043.

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

Sun, C., Sun, M., Chen, H.-T., 2021. Hohonet: 360 indoor holistic understanding with latent horizontal features. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2573–2582.

The CGAL Project, 2022. CGAL User and Reference Manual, fifth.fourth ed. CGAL Editorial Board, URL: https://doc.cgal.org/5.4/Manual/packages.html.

Wang, X., Marcotte, R.J., Olson, E., 2019. GLFP: Global localization from a floor plan. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, pp. 1627–1632.

Wold, S., Esbensen, K., Geladi, P., 1987. Principal component analysis. Chemometr. Intell. Lab. Syst. 2 (1–3), 37–52.

Yang, S.-T., Wang, F.-E., Peng, C.-H., Wonka, P., Sun, M., Chu, H.-K., 2019. Dula-net: A dual-projection network for estimating room layouts from a single rgb panorama. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3363–3372.

Yu, F., Koltun, V., Funkhouser, T., 2017. Dilated residual networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 472–480.

Yu, Z., Zheng, J., Lian, D., Zhou, Z., Gao, S., 2019. Single-image piece-wise planar 3d reconstruction via associative embedding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1029–1037.

Ziran, Z., Marinai, S., 2018. Object detection in floor plan images. In: IAPR Workshop on Artificial Neural Networks in Pattern Recognition. Springer, pp. 383–394.

Zou, C., Colburn, A., Shan, Q., Hoiem, D., 2018. Layoutnet: Reconstructing the 3d room layout from a single rgb image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2051–2059.