ISPRS Journal of Photogrammetry and Remote Sensing 170 (2020) 29–44

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

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

# Robust 3D reconstruction of building surfaces from point clouds based on structural and closed constraints

Senyuan Wang [a,f], Guorong Cai [a,b,*], Ming Cheng [c], José Marcato Junior [d], Shangfeng Huang [a], Zongyue Wang [a], Songzhi Su [c,*], Jonathan Li [b,c,e]

[a] School of Computer Engineering, Jimei University, Xiamen, FJ 361021, China
[b] Fujian Collaborative Innovation Center for Big Data Applications in Governments, Fuzhou University, Fuzhou, FJ 350003, China
[c] School of Informatics, Xiamen University, Xiamen, FJ 3610005, China
[d] Faculty of Engineering, Architecture and Urbanism and Geography, Federal University of Mato Grosso do Sul, Brazil
[e] Department of Geography and Environmental Management and Department of Systems Design Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada
[f] School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, HB 430079, China

ARTICLE INFO

*Keywords:*
3D reconstruction
Building surface reconstruction
Closed-constraints
LiDAR point clouds

ABSTRACT

The reconstruction of buildings using inhomogeneous and unstructured point clouds is a challenging task for photogrammetry and computer vision research communities. A new approach for 3D building surface modeling, based on closed constraints, is proposed. First, a region growth algorithm is applied to fit the input point clouds by a set of candidate planes. Then, additional candidate planes are generated from the initial planes according to a rigid transformation followed by expanding the original primitive set to the candidate model set through generation rules. Furthermore, an energy function is employed to combine the data fitting errors with the structural constraints at the model selection stage. Finally, the 3D building surface model is generated from the candidate set through energy minimization. More precisely speaking, we adopt the surface optimization scheme that enforces the 3D polygonal surfaces of the building to be consistent with a priori geometric structures. Our approach was assessed using multi-source datasets with different densities, noise levels covering diverse and complex structures. The experimental results demonstrated that the proposed approach achieves better accuracy and robustness than those of several state-of-the-art methods.

## 1. Introduction

The surface reconstruction from 3D point clouds is crucial for several applications (Berger et al., 2017; Tachella et al., 2019; Li et al., 2019), including those related to virtual reality (Bruno et al., 2010), smart cities (Yang and Lee, 2017) and robotics (Rajput et al., 2018). Building surface reconstruction is a fundamental task to achieve large-scale 3D city reconstruction. Notably, the most challenging task is to propose a robust algorithm for 3D surface reconstruction to generate reliable building surfaces from multiple-sources data with different sparsity, varying noise levels, and complex structures. To this end, a large amount of research has focused on building surface reconstruction using various methods, including image-based (Musialski et al., 2012; Nan et al., 2015; Wang et al., 2015), aerial LiDAR-based (Verma et al., 2006; Jarzkabek-Rychard and Borkowski, 2016; Wu et al., 2017) and ground LiDAR-based (Chen and Chen, 2008; Wan and Sharf, 2012). Reconstruction of building surfaces in complex real-world scenes poses specific challenges due to various geometric shapes and occlusions. Consequently, automated reconstruction of 3D building surfaces from point clouds yet remained an open challenge (Ochmann et al., 2019; Berger et al., 2017). Many methods have been proposed, including those based on implicit methods (Hoppe et al., 1992, 1994), moving least-squares fitting method (Alexa et al., 2003), Multilevel Partition of Unity (Ohtake et al., 2003), Poisson surface reconstruction method (Kazhdan et al., 2006; Alliez et al., 2007), and Bayesian surface reconstruction methods (Jenke et al., 2006; Diebel et al., 2006). These implicit methods sometimes tend to smooth the data. In order to produce a sharper surface, unlike the $l_1$- or $l_2$- norm in many methods, Li et al. (2018) proposed a new minimization method by using $l_0$ gradient. Unfortunately, most of these approaches do not cope with poor quality point clouds. To overcome this, Ganapathi-Subramanian et al. (2018) used structure-aware shape templates to guide the generation of initial

* Corresponding authors at: College of Computer Engineering, Jimei University, Xiamen, FJ 361021, China (G. Cai).
*E-mail addresses:* guorongcai.jmu@gmail.com (G. Cai), ssz@xmu.edu.cn (S. Su).

https://doi.org/10.1016/j.isprsjprs.2020.09.004
Received 30 January 2020; Received in revised form 1 September 2020; Accepted 4 September 2020
Available online 14 October 2020

primitives. However, a limitation occurs due to the variety of shape templates, and it becomes a challenge to generate new shapes. Li et al. (2017) and Mo et al. (2019b) adopted neural network architecture to capture the layered structure of the reconstructed object to achieve finer geometric perception of a complex structure. Moreover, some valuable scene priors were used for building surface reconstruction. For example, it is beneficial to reconstruct the building surface fully by extracting high-level global scene features instead of local details (Yumer and Kara, 2012; Oesau et al., 2014). In most urban scenes, buildings can be conceived as a composition of surface structures and their relationships. As such, graph neural networks that are able to describe relationships on graphs of 3D point clouds has attracted widespread attention. Mo et al. (2019a) presented a hierarchical graph network for concurrently combining both part geometry and inter-part relations during network training. Similarly, Li et al. (2018b) used graph neural networks to represent probabilistic dependencies among graphs nodes and edges, which can capture structure and attributes. These methods have certain guiding significance for the identification of relations between primitives.

This paper, presents a new approach to building surface reconstruction with a focus on the closed constraints when selecting global inter-plane relations of primitives to obtain the watertight surface model of regularized primitives. More precisely, we developed an energy minimization strategy, which taking into account structural closure constraints and the adjacency of candidate planes. The closed constraints are applied to ensure the coincidence of the intersecting edges of the adjacent planar surfaces.

The rest of this paper is organized as follows: Section 2 briefly reviews related studies in surface reconstruction. Section 3 presents the motivation to conduct this work. Section 4 gives specific implementation steps for the proposed algorithm, including how a candidate was generated, as well as the closed and regular arrangements of planes selection. Section 5 presents and discusses the experimental results and compares the performance of our method with that of several existing methods. Section 6 concludes of the paper.

## 2. Related works

The development of 3D sensing equipment has facilitated the acquisition of point clouds. Reconstruction of building models from 3D point cloud has become the main research content in computer graphics and photography, being implemented a lot of research in the past few decades. The polygonal surface reconstruction aims to obtain dense and complete surfaces. According to the difference of reconstruction strategies, the existing surface reconstruction methods of buildings from point clouds mainly include two categories: data-driven and structure-driven.

As for data-driven reconstruction methods, the most popular technique fits geometry and identifies higher-level intrinsic structural relationships between them. The extraction of initial primitives is mainly to restore the real object model from the point clouds with noise and outliers. The milestone method is based on Random Sample Consensus (RANSAC) (Schnabel et al., 2007), which is an efficient model fitting algorithm, as well as its variants, such as variational surface approximation (Li et al., 2009), Groupsac (Ni et al., 2009), PROSAC (Chum and Matas, 2005) and Multi-Structure Estimation (Wong et al., 2011). The sampling-based approachs are popular because of simplicity, scalability, and probability guarantees. However, the performance of scene-level structures is relied on the iterative selection of inner point sets from the data to estimate the model, in which the number of iterations of the algorithm is large and unstable. The GlobFit framework (Li et al., 2011) optimizes the data fitting based on the intrinsic relationships of the original primitives (obtained by RANSAC), further identifying global relationships. However, if the extracted primitives are not accurate, influenced by the of noise points, the detection of the wrong relationship between primitives can be generated. Based on RANSAC, Jenke et al. (2008) proposed an algorithm to extract the basic geometric model and candidate contour points to provide boundary contour through an optimization function. By subdividing the surface reconstruction into sub-problems of each primitive, the error boundary extraction of one primitive does not affect the surface reconstruction of the rest, solving the problem of fails due to local errors occurred in the global optimization method.

Compared to approaches that focus on obtaining dense polygonal surfaces, the exploitation of high-level plane has attracted researchers' attention. Arikan et al. (2013) proposed a semiautomatic polygon surface reconstruction method, which refines the initially fitted primitives through manual interaction. Considering the adjacency of the polygon, Chen and Chen (2008) connected the outer boundary points to perform polygon fitting. This method is feasible when studying a model refined against poor-quality data because the intersection of adjacent planes can be inferred from the boundary of the fitting plane, thereby restoring the structure of the missing portion. Taking the data as the research object, PEARL approach (Isack and Boykov, 2012) mathematically models the fitting problem and uses the energy function for the optimization task. However, broader inter-primitive relations are needed. Besides, the reconstruction can be transformed into a volume reorganization of locally block-level structures from the perspective of combination. For example, Lin et al. (2013) reconstructed the rough building model by decomposing the point cloud into block levels and piecewise fitting faces.

Similarly, Yi et al. (2017) projected the boundary points dividing the building into sections regarding the distribution histogram of the projected points. However, the segmented block structure fails when a failure occurs in the boundary extraction caused by noise interference. In this case, structural constraints based on primitive extraction become an alternative to improve the reconstruction accuracy.

The structure-driven reconstruction methods consider the repetitive and symmetrical structures of buildings. For example, Smart-Boxes (Nan et al., 2010) focuses on the reconstruction of tall buildings with many repetitive structural elements. The structural elements are selected by the user, matching with similar shapes in the model library. Then, the matched models are integrated to generate regular models, as presented by Pauly et al. (2008) and Gal et al. (2007). Moreover, symmetry or repetition is the most common type of structure, and their automatic detections are implemented in 2D by Furukawa et al. (2009), based on projection geometry (Hartley and Zisserman, 2003). Li et al. (2011) explored the global structural relationships of primitives to fit the scan data better, however not handling the data that completely lacks a face. Chen et al. (2017) divided the roof according to the topology rules. Then, the extraction of the original boundary using Voronoi graph-based was conducted (Amenta et al., 1998), and used to generate the lightweight building model. It is worth noting that the original shape and its spatial relationship are indispensable in the existing primitive-based approach. Monszpart et al. (2015) proposed an artificial scene reconstruction algorithm based on the regular set, which uses an iterative method to simultaneously extract primitives and local internal relationships, and generate a small main structure of the model. However, the primary purpose of these viewpoints is to infer the symmetry, vertical, and other structural relationships between the primitives, and fail to guarantee the closure of the model boundaries. Nan and Wonka (2017) obtained watertight reconstructions surface models of the buildings based on an approach that is effective in dealing with noise, outliers, and missing data. However, small principal structures are easily missed, and it is only applied to the reconstruction of a fully closed model. In recent years, there are methods on generative models for 3D shapes, for example, reconstruction methods based on point clouds (Achlioptas et al., 2019; Li et al., 2018a; Fan et al., 2017), modeling methods based on multi-scale depth maps (Arsalan Soltani et al., 2017), reconstruction methods based on surface meshing (Groueix et al., 2018) and local shape synthesis (Kalojanov et al., 2019), and so on. These methods focus more on the basic geometric structure generation rather than the overall structure. Recently, Xia et al. (2020) investigated the

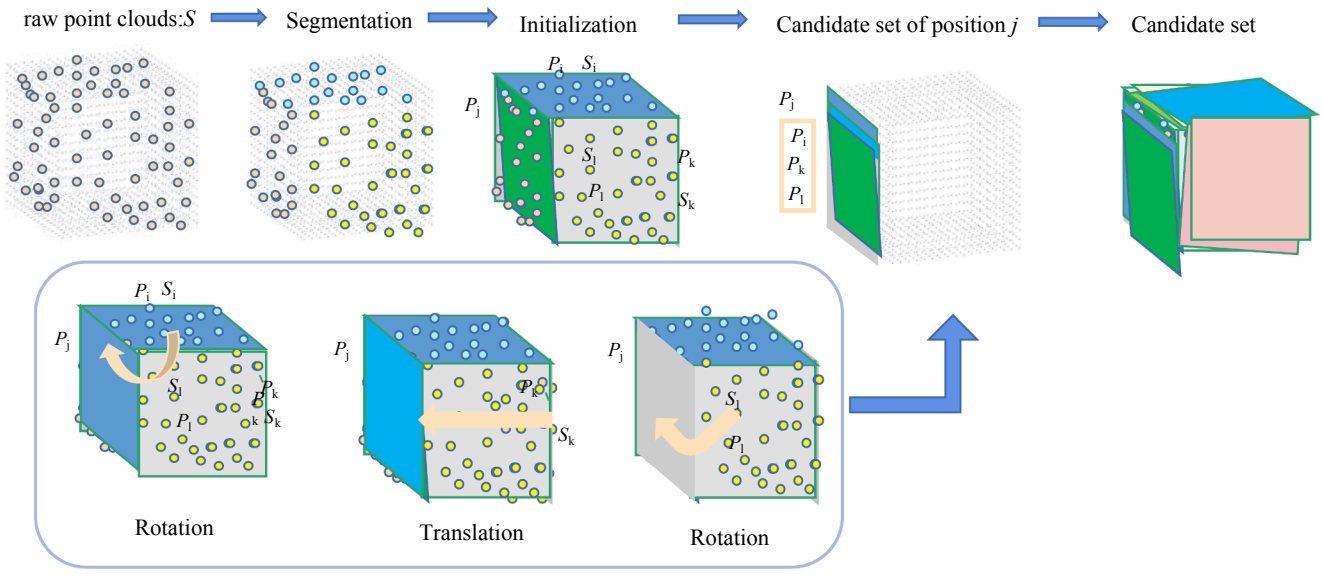



**Fig. 1.** The pipeline of candidate generation.

latest processing techniques regarding geometric primitives, emphasizing that primitive extraction methods can still achieve the globally optimal results effectively, even with poor Lidar data quality.

The algorithms mentioned above ignore the boundary closure problem during the optimization reconstruction process, focusing only on the local fitting or on inferring the internal relationships of the primitives. The proposed algorithm guarantees the closeness of the boundary while inferring the relationship between the surface sets to generate a more watertight high-precision model.

## 3. Motivation

Structural relationships in real scenes are often biased or even wrong due to the lack of points and the presence of noise. According to the Manhattan hypothesis, the building is mainly composed of plane parts in most urban human-made building scenes. Typical planar relationships between them include parallelism, coplanarity, orthogonality, and symmetry, etc. Therefore, we first speculate on possible relationships and generate candidate sets that are likely to be close to the real model based on hypothetical relationships. In other words, the candidate primitive planes are first expanded by adding hypothesized planes. Secondly, the regular arrangements of planes (RAP) extraction are transformed into a selection problem.

An effective way to achieve this premise is first to find potential inter-plane relationships as much as possible based on the initial fitting primitives. Then extract the proper intrinsic relationship of the primitives according to the constraints. As for the constraint rules, we consider three aspects: (1) data cost: the fitting residual between the original data points and the primitives; (2) irregularity cost: the gap between the structural relationship between the continuous primitives and the prior prediction. Besides, the additional smoothing term is to ensure the smoothness of the segmentation.

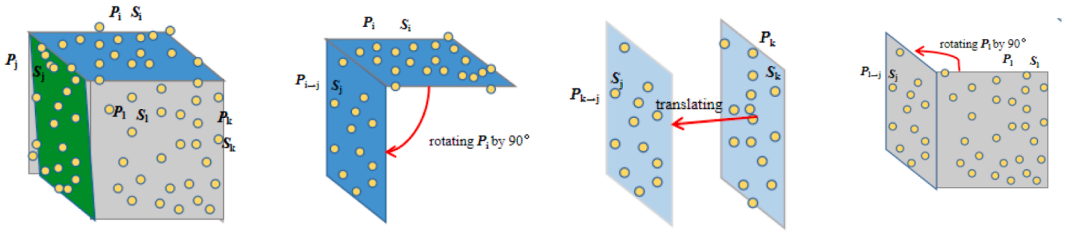
The rule constraints of the structure are generated from a global perspective based on the reconstruction algorithm of the regular set, considering the topological relationship between the faces. However, the discontinuity of reconstructed models may occur due to occlusion. RAPter method (Monszpart et al., 2015) did not consider this.

Strategies on boundary closure constraints of adjacent structures to optimize the accuracy of the model fitting are needed. Our method, Clow_Rapter, considers the original point cloud as input, which may contain outliers, noise, and missing data. The output of the proposed method is the permutation of primitives selected for optimization constraints and their intrinsic relationships.

The algorithm is implemented in three steps: (i) the original point cloud is initially fitted; (ii) the additional candidate planes are created using the initial planes by rigid transformation, and the original primitive set $P$ is expanded to the candidate model set $\tilde{P}$ according to the generation rules; (iii) an optimization formula is developed to select Clow_RAP from the expanded candidate set, taking into account the structural constraints of the global perspective and the model closure in the Clow_RAP selection stage, which is the most critical.

## 4. Method

The proposed method aims to reconstruct the closure and regular arrangements of the planes from a raw point cloud of the scanned buildings, where the regular arrangement refers to the orderly combination of primitives with a certain structural relationship. The proposed




(a) Initialization (b) $P_i$ rotates to $\boldsymbol{S}_j$ (c) $P_k$ translates to $\boldsymbol{S}_j$ (d) $P_l$ rotates to $\boldsymbol{S}_j$

**Fig. 2.** Graphical representation of candidate generation.

method considers three main stages:

(1) **Initialization:** First, the raw point cloud $\boldsymbol{S}$ is segmented by a region growing algorithm into a set of patches $\{\boldsymbol{S}_j\}$ according to the normal consistency, where $j$ is an index of each block cloud. Then, we calculate the fitting plane $P_j$ of each patches $\boldsymbol{S}_j$ using the least-squares method. Thus, the initial set of primitive planes $\boldsymbol{P}$: $\{P_j\}$ are obtained;

(2) **Candidate generation:** Since the original primitive $\boldsymbol{P}$:$\{P_j\}$, generated by local fitting, is susceptible to outliers, it is difficult to detect the structural relationship between these disordered local primitives robustly. In Fig. 1, a 3D cube is considered as a toy example. The initial fitting planes of the patches $\boldsymbol{S}_i$, $\boldsymbol{S}_j$, $\boldsymbol{S}_k$, $\boldsymbol{S}_l$ in initialization are denoted as $P_i$, $P_j$, $P_k$, $P_l$, respectively. The point cloud at position $j$ is scattered due to the influence of noise. The points at this location are supposed to be coplanar. As shown in Fig. 2(a), there are some points deviate from the plane, generating inclined planes, such as $P_j$. From a global perspective, the patches $\{\boldsymbol{S}_j\}$ may be better fitted by the plane model of other locations through a rigid transformation. As shown in Fig. 1, the primitive $\{P_i\}$ which is a primitive that is initially fitted by the patches $\{\boldsymbol{S}_i\}$ is rotated by 90° to the geometric center of the patches $\{\boldsymbol{S}_j\}$. Similarly, $P_l$ has been rotated to the center of $\{\boldsymbol{S}_j\}$. Primitive $P_k$ is parallel to $P_j$, and then $P_k$ is translated to a certain distance to the center of $\{\boldsymbol{S}_j\}$. As a result, the fitting model of the patches $\{\boldsymbol{S}_j\}$ has a set of new candidate sets $\{P_i, P_l, P_k\}$. Generally speaking, the candidate sets can be expressed as $\{P_{i\to j}, \ldots, i, j=1, 2,\ldots\}$, where $P_{i\to j}$ indicate primitives that are transformed from $\{\boldsymbol{S}_i\}$. The primitive $P_i$ fitted by the patches $\{\boldsymbol{S}_i\}$ is transformed to the center of $\{\boldsymbol{S}_j\}$, and this primitive is denoted as $P_{i\to j}$. To highlight that, $P_j$ is obtained by the transformation of $P_i$, which is denoted as $P_{i\to j}$. According to the above rules, more primitives are generated for each patch. Therefore, we gain fitting models $\tilde{P} = \{P_{i\to j}, i, j=1, 2, \ldots\}$, which are suitable for the raw point cloud. Although these primitives contribute to recovering the structures with missing data or excessive noise, most of them are eventually discarded.

(3) **Closed regular arrangements of planes selection:** We select closed sequences with regular structure from the extended set of primitives $\tilde{P}$. The binary variable $\chi_{i\to j} \in \{0,1\}$ indicates whether the candidate plane $P_{i\to j}$ is selected as the final model. As shown in Fig. 2, the candidate model of the patch $\{\boldsymbol{S}_j\}$ is obtained by the rotation and the translation transformation of $P_i$, $P_k$, and $P_l$. As shown in Fig. 2(b), $P_i$ is rotated by 90°, and then translated to the center of the patches $\{\boldsymbol{S}_j\}$ to obtain $\{P_{i\to j}\}$. Similarly, $\{P_{k\to j}\}$ and $\{P_{l\to j}\}$ are candidate models for $\{\boldsymbol{S}_j\}$. If $\{P_{k\to j}\}$ is selected as the effective fitting primitive of the patch $\{\boldsymbol{S}_j\}$, then $\chi_{i\to j} = 1$, otherwise, $\chi_{i\to j} = 0$.

The binary variable $[\chi_{i\to j}, \ldots], i = 1, 2, \ldots$ represents the final set of valid models, called the regular set. The regular set needs to be selected according to the structural similarity grouping due to the complexity of the topology of the building. Therefore, the indicator variable $\chi_{i\to j} \in \{0,1\}$ is introduced, and $\chi_i$ represents the original primitive of the transformation in a set of models. So the value of the indicator variable $\chi_i$ is $\max_j \chi_{i\to j}$. This algorithm transforms the selection problem of the regular set into the minimum optimization problem of the energy function, which can be described by Eq. (1).

$$\{\chi_i\}, \{\chi_{i\to j}\} = argmin\ E := \underset{\{\chi_i\}\{\chi_{i\to j}\}}{\lambda E_{data}} + (1-\lambda)E_{irr^*} + E_{spat} \tag{1}$$

where $E_{\text{data}}$ is the data item, $E_{\text{irr}^*}$ is the non-regular term, and $E_{\text{spat}}$ is the spatial smoothing term, with $\sum_j(\chi_{i\to j}\chi_i - \chi_{i\to j}) \geqslant 0$. It is guaranteed that when $\{P_{i\to j}\}$ is selected as the effective fitting primitive of $\{\boldsymbol{S}_j\}$, the primitive $\{\boldsymbol{S}_i\}$ of the set of models also needs to be valid. $\lambda \in \{0,1\}$, being close to 1 when the input data has high-precision or is pre-processed. On the other hand, when the data contains a lot of noise or data missing, we can choose a small value of $\lambda$. In general, the value of $\lambda$ is set to be 0.5.

### 4.1. Structural constraint

#### 4.1.1. Data

The purpose of the data cost is to calculate the fitting residuals between the original point cloud $\{P_h\}$ in $\{\boldsymbol{S}_j\}$ and its fitting primitives $\{P_{i\to j}\}$. In the initial fitting phase, the regular set based reconstruction algorithm use Principal Component Analysis (PCA) (Maćkiewicz and Ratajczak, 1993) to estimate the sum of the fitted distance errors, as presented in Eqs. (2) and (3).

$$E_{data} := \sum_j \sum_i \chi_{i\to j} E_d(P_{i\to j}, S_j) \tag{2}$$

$$E_d(P_{i\to j}\ ,\ S_j) = \frac{1}{\left|\{p_h \in S_j\}\right|} \sum_{p_h \in S_j} d(p_h, p_{i\to j})^2 \tag{3}$$

where the binary variable $\chi_{i\to j}$ indicates whether a candidate plane $\{P_{i\to j}\}$ is selected as the fitting model of the patches $\{\boldsymbol{S}_j\}$. Due to the different sizes of point cloud blocks $\{P_h\} \in \boldsymbol{S}_j$, $E_{\text{d}}$ normalizes the fitting error of each patch $\{P_{i\to j}\}$.

#### 4.1.2. Irregularity

After ensuring the accuracy of the primitive fitting, the structural relationship among the primitives also needs to be maintained. Similar to the Manhattan-World assumption, the parallel planar arrangement is the most common rule. The non-regular term $E_{\text{irr}^*}$ evaluates the degree of matching between primitive models. In our algorithm, each primitive $P_{\text{i}}$ is transformed into the position of the other patches. Without loss of generality, $P_i$ fitted by $\{S_i\}$ is represented as $P_{i\to i}$. The irregularity term is then defined in Eq. (4).

$$E_{irr^*} := \sum_{i,k} \chi_i \chi_k Irr(p_{i\to i}, p_{k\to k}) \tag{4}$$

where $Irr(P_{i\to i}, P_{k\to k})$ is an irregularity function, $Irr(P_{i\to i}, P_{k\to k}) := f(\angle(n_i, n_k))$. The $f(\cdot)$ function is the mathematical formula $f(x) := 1-\exp(-\delta x)$. $\angle(n_i, n_k)$ represents the angle between the normal vector of primitive $P_{i\to i}$ and the normal vector of $P_{k\to k}$ normal vector. $\chi_i$ indicates whether the model $P_{i\to i}$ is selected or not. Namely $\chi_i = 1$ means $P_{i\to i}$ is selected, otherwise $\chi_i = 0$.

#### 4.1.3. Spatial smoothness

Spatial smoothness term merges the patches of adjacent structures in the same direction to be merged into coplanar patches. The two patches $\boldsymbol{S}_j$ and $\boldsymbol{S}_l$ are merged if a minimum distance occurs between their points, and also if there is a close direction vector between the two patches, as presented in Eq. (5).

$$E_{spat} := \sum_{j,i,l,k; j\neq l} \chi_{i\to j}\chi_{k\to l} neigh(S_j, S_l) C_{spat} \tag{5}$$

where $C_{\text{spat}}$ is a fixed space penalty, which is defined as $C_{\text{spat}} = (1-\lambda)/10$. The $neigh(\cdot)$ function measures the proximity of the distance and direction of two patches in Eq. (6).

$$neigh(S_j, S_l) := I(\min_{p_g \in S_j; p_k \in S_l} \|p_g - p_k\| < 2\rho) I(\angle(n(S_j), n(S_l)) < \tau) \tag{6}$$

where $I(\cdot)$ is a binary function, note that $I(x)=1$ when the conditions "$x$" in parentheses is satisfied, otherwise $I(x)=0$. The $neigh(\cdot)$ function measures the proximity of the distance and the direction of two patches. In particular, the patches are considered adjacent if the distance between patches $\boldsymbol{S}_j$ and $\boldsymbol{S}_1$ is less than $2\rho$, and the normal vector direction angle is less than $\tau$. The distance between the patches $\boldsymbol{S}_j$ and $\boldsymbol{S}_l$ is

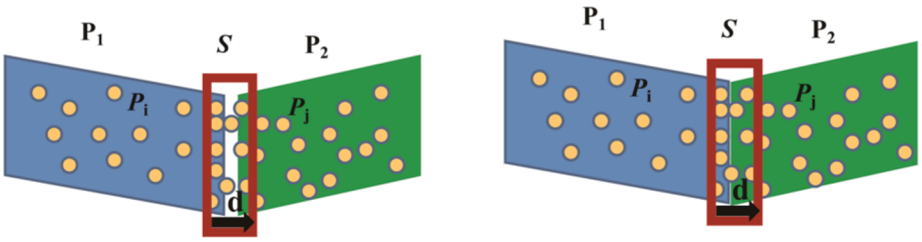



**Fig. 3.** The closed constraint of adjacent faces. (a) The gap between adjacent faces; (b) Selection of a closed plane pair by candidate constraint in the candidate set.

calculated by the minimum distance between the two points $P_g$ ($P_g \in \boldsymbol{S}_j$) and $P_k$ ($P_k \in \boldsymbol{S}_1$).

### 4.2. Closed constraint

The extraction algorithm of regularly arranged planes mainly focuses on obtaining closed and regular sequences of primitives, ignoring the boundary connection of adjacent faces. Most critically, high-quality closed models are obtained based on global structural relationships. To achieve the closure of adjacent planes, we transform the closure of the gap between the faces into the constraint adjacency. If the adjacent faces intersect, the original points at the intersecting edges must be close enough. The algorithm mainly consists of two stages.

(1) Determine each pair of adjacent faces:

The plane $\boldsymbol{P}_1$ is obtained by fitting the patch $\boldsymbol{S}_1 = \{p_i, i = 1,2,3,...\}$. Similarly, $\boldsymbol{P}_2$ is obtained by fitting the patch $\boldsymbol{S}_2 = \{p_j, j = 1,2,3,...\}$. If $\boldsymbol{P}_1$ is adjacent to $\boldsymbol{P}_2$, it need satisfy:

If the distance of patches $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ is close enough, it means that the points contained in patches $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ are close to each other. In our assumption, we define the points set $\boldsymbol{S}$, where $\boldsymbol{S} = \{(p_i,p_j) \; ||p_i\text{-}p_j|| < \mathrm{d}^*\}$, $i$, $j = 1,2,3,\ldots$, $p_i \in \boldsymbol{S}_1$, $p_j \in \boldsymbol{S}_2$. $\boldsymbol{S}$, as a non-empty set, contains points in two point cloud blocks whose distance from each other is less than the threshold, as shown in Fig. 3(a).

(2) The closed constraint of adjacent faces:

Given adjacent and intersect planes $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$, they must satisfy the following conditions: the distance between points $\boldsymbol{p} \in$ S from the $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ is sufficiently close, and the sum of the distances to $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ should be as close as possible within a distance threshold. The energy minimum with closed constraints selects the final model set (see Fig. 3(b)). This process can be described by Eq. (7).

$$E_{closure} = dis(\boldsymbol{S}, P_1) + dis(\boldsymbol{S}, P_2) \tag{7}$$

By adding this closing constraint into Eq. (1), we obtain the updated energy function (Eq. (8)).

**Table 1**
Statistics of processed scenes: the number of original input points (#points), and # point resolution is the average distance between measurement points.

| Scene | #points | #point resolution/ m | Scene | #points | #point resolution/ m |
|---|---|---|---|---|---|
| *Container* | 81 k | 0.08 | *Lans* | 126 k | 0.004 |
| *Contain_unit* | 162 k | 0.08 | *Room* | 586 k | 0.004 |
| *Tower* | 65 k | 0.4 | *Xiaowu* | 100 k | 0.02 |
| *Bungalow* | 71 k | 0.3 | *Polyfit_4e* | 118 k | 0.08 |
| *Empire* | 1.2 M | 0.0025 | *Bottom* | 159 k | 0.3 |
| *Boxunion_noise* | 100 k | 0.08 | *Apartment* | 105 k | 0.8 |
| *Euler* | 1.22 M | 0.02 | *Nola* | 78 k | 0.05 |
| *Boxunion* | 739 k | 0.004 | *Notre-Dame de Paris* | 78 k | 0.02 |

$$\{\chi_i\}, \{\chi_{i\to j}\} = argmin\ E := \underset{\{\chi_i\}\{\chi_{i\to j}\}}{\lambda E_{data}} + (1-\lambda)E_{irr^*} + E_{spat} + E_{closure} \tag{8}$$

The energy function contains the data term $E_{\text{data}}$, the non-regular term $E_{\text{irr}^*}$, and the spatial smoothing term $E_{\text{spat}}$. The energy formulation in Eq. (8) contains the data fitting error term $E_{\text{data}}$, the irregularity term $E_{\text{irr}^*}$, the spatial smoothness term $E_{\text{spat}}$ and the closed constraint $E_{\text{closeure}}$. The data fitting error term $E_{\text{data}}$, the irregularity term $E_{\text{irr}^*}$, the spatial smoothness term $E_{\text{spat}}$ are described in detail in Eq. (1).

### 4.3. Optimization

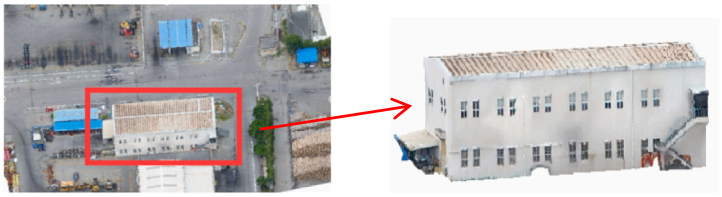
The purpose of the optimization is to search the final model from the expanded candidate set based on the energy function minimization. We used the CoinBonmin solver (Monszpart et al., 2015) for the task of optimization. Bonmin (Basic Open-source Nonlinear Mixed INteger programming) is an open-source code for solving general MINLP (Mixed Integer NonLinear Programming) problems. The web address of the open-source code is https://projects.coin-or.org/svn/Bonmin/stable/1.5. We adopted the strategy from coarse to fine to improve the optimization efficiency of the algorithm. Specifically, by setting a threshold that supports the number of plane points, the planes that meet the threshold range are selected in batches for optimization. For example, if the threshold is set to be 800, there must be at least 800 points associated with the plane that related to the optimization. Then, the next level candidate set is introduced according to the previous extraction process. When selecting primitives in each iteration, the sum of reconstruction costs is calculated. In the next iteration, the face set with the smallest total cost is selected. The specific algorithm is implemented as Algorithm 1.

**Algorithm 1.** (*Algorithm of Closed-Regular Arrangements of Planes (Clow_Rapter)*)

**Input**:patches$\{\boldsymbol{S_j}\}\in\boldsymbol{S}$, local primitives$\{P_{i\to i}\in\boldsymbol{P}\}$, area threshold $\alpha$
**Output**:Clow_RAP set $\boldsymbol{P}^*=\{P_{i\to j}\}$, closed relationship $\{<P_{i\to j}, P_{i\to k}, relation>\}$
/// (1) Initialization
1: $\boldsymbol{P_0}:=\{\boldsymbol{P}_{j\to j}\}\in\boldsymbol{P}$, $\boldsymbol{P}^* := \Phi$, $\boldsymbol{S} == \boldsymbol{\Phi}$, $\boldsymbol{P}_* = \boldsymbol{\Phi}$ // Initialize set of selected candidates $P_0$, points set of adjacent plane pair $S$, the set of adjacent plane pair $P_*$
2: population=num // faces require that there are enough close pairs of points on both faces
3: $\alpha$:= $\text{area}_{\max}/\text{area}_{\min}$ $\rho$
/// (2) Candidate generation
4: **while** $\boldsymbol{P}^0 \neq \emptyset$ **do**
5: $\boldsymbol{P}_\alpha$:=$\{\boldsymbol{P}_{j\to j}$, s.t. $\forall j$ with area$(\boldsymbol{P}_{j\to j}) > \alpha\}\in\boldsymbol{P_0}$
6: $\boldsymbol{P_0}$:=$\boldsymbol{P_0}\backslash\boldsymbol{P}_\alpha$
// expand from a set of primitives
7: $\tilde{\boldsymbol{P}}$:=$\boldsymbol{P}_\alpha\cup$Enrich$(\boldsymbol{P}_\alpha, \boldsymbol{P}^*)\cup$Enrich$(\boldsymbol{P}_\alpha, \boldsymbol{P}_\alpha)$
/// (3) Clow_RAP Selection
// Closed constraint
8: if distance$(p_i\in\tilde{\boldsymbol{P}}(i), p_j\in\tilde{\boldsymbol{P}}(j))<\mathrm{d}^*$) then // If the distance between the points in the two patches is less than the threshold d*
9: $\boldsymbol{S}$=$\boldsymbol{S}$+$(p_i, p_j)$ // Add point pairs to collection **S**
10:end if
11: if $\boldsymbol{S}$.size() > population then
12: $\tilde{\boldsymbol{P}}(i)$ adjacent to $\tilde{\boldsymbol{P}}(j)$
13: $\boldsymbol{P}_*$=$(\tilde{\boldsymbol{P}}(i),\tilde{\boldsymbol{P}}(j))$

(*continued on next page*)

(a) *Container* (b) *Tower*



(c) *Bungalow*

**Fig. 4.** Aerial images.

(*continued*)

14:end if
15:**return** $P_*$
// Select Clow_RAP
16: $P^*$:=$P^* \cup P_* \cup \{P^* \subseteq \tilde{P}\}$
17: control the number of candidate set generations, decrease the threshold $\alpha = \alpha/2$ // Iteration
18: end while
19: **return** $\tilde{P}^*$

## 5. Algorithm implementation and assessment

### 5.1. Experimental setup

We implemented our method using C++ under the ubuntu system. The number of candidate sets is controlled by setting a threshold during the candidate set generation stage. The distance threshold in the experiment is based on the point density presented in Table 1.

### 5.2. Data_sets

In the experiment, the data sets include an aerial image-based point

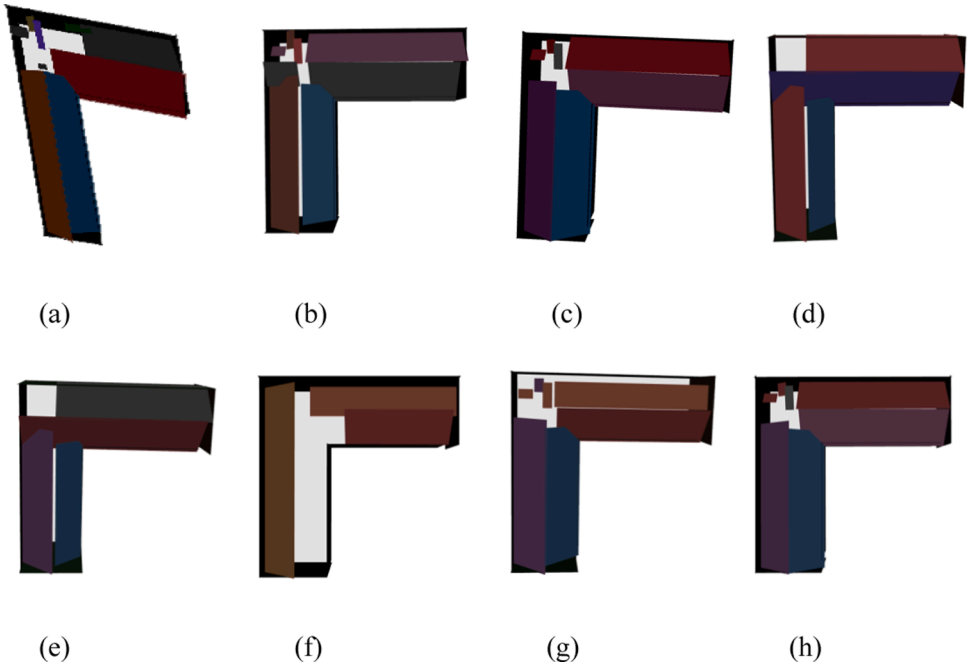



**Fig. 5.** The final model generated by different input parameters (a) to (h) as listed in Table 2: By adjusting the values of pairwise (*pw*) and poplimit (*pl*), the models with different degrees of detail are generated.

**Table 2**
The input parameters corresponding to the reconstruction results shown in Fig. 5 (a)-(h).

| | Result | -s/m | -al/° | -pw | -pl | -it/times | -t |
|---|---|---|---|---|---|---|---|
| a | 14 | 0.02 | 15 | 0.01 | 1 | 20 | 16 |
| b | 14 | 0.02 | 15 | 0.01 | 3 | 20 | 16 |
| c | 14 | 0.02 | 15 | 0.01 | 2 | 20 | 16 |
| d | 10 | 0.02 | 15 | 0.01 | 8 | 20 | 16 |
| e | 10 | 0.02 | 15 | 0.01 | 10 | 20 | 16 |
| f | 9 | 0.02 | 15 | 0.01 | 20 | 20 | 16 |
| g | 13 | 0.02 | 15 | 0.05 | 3 | 20 | 16 |
| h | 13 | 0.02 | 15 | 0.015 | 2 | 20 | 16 |

cloud and a LiDAR point cloud. The image-based data such as *Container*, *Contain_unit*, *Tower*, and *Bungalow* are from Jimei University, China. The structure of the *Bungalow* is more complicated because the point clouds are from the reconstructed results of aerial images, which were acquired from the suburbs of Xiamen City, as shown in Fig. 4. *Empire* and *Box-union_noise* are LiDAR point clouds with noise. *Empires* have a relatively low noise level and complicated structure, which is derived from Lafarge and Alliez (2013). *Boxunion* is a courtesy from the Intelligent Geometry Processing Laboratory at the University of London (Guerrero et al., 2018), and *Boxunion_noise* (LiDAR) contains Gaussian noise. Besides, the point clouds of complex and diverse structures are also provided. For example, *Euler* (LiDAR) is from Oesau et al. (2014), *Lans* is from Lafarge

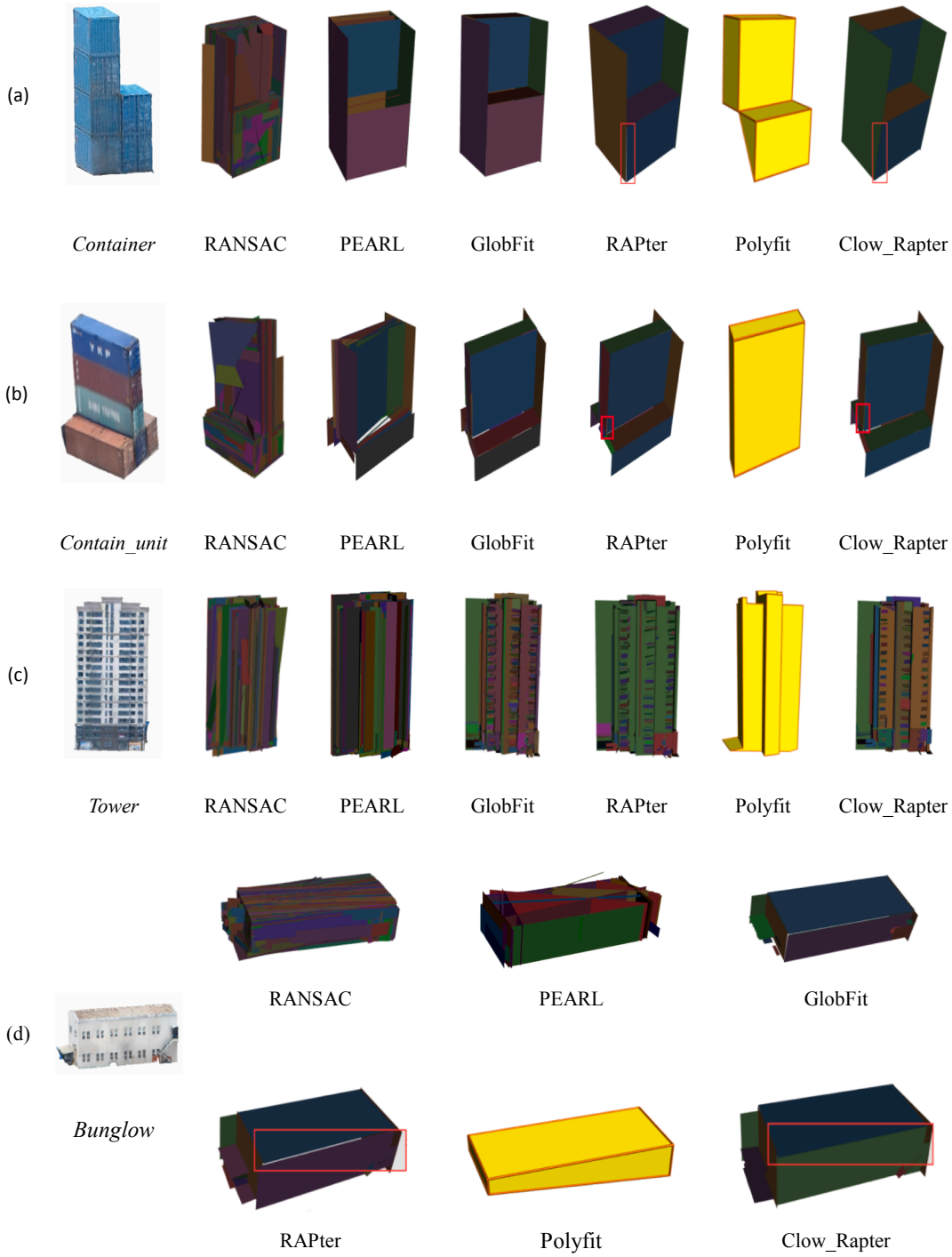



**Fig. 6.** Reconstruction of a set of buildings from 3D-imaging cloud points, which corrupted by the inhomogeneous and unstructured point distribution.

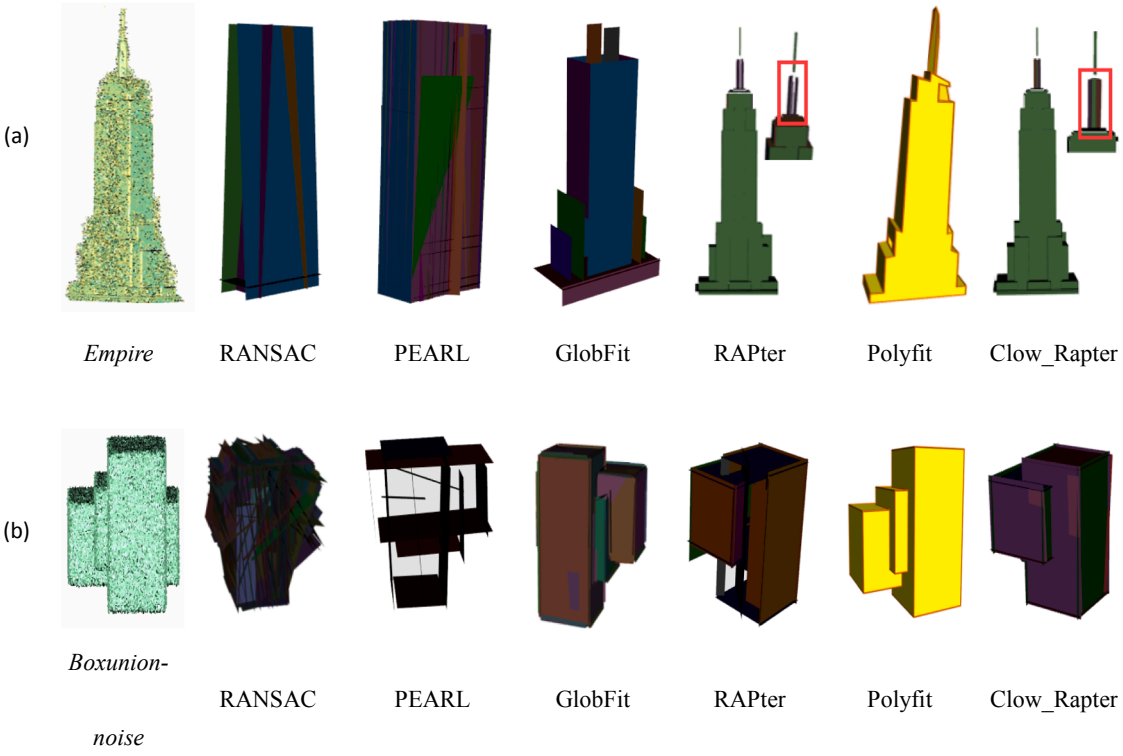



**Fig. 7.** Reconstruction results of the *Empire* and *Boxunion_noise* data sets, which are corrupted by outliers.

and Alliez (2013), and *Nola* is derived from Zheng et al. (2010). *Notre-Dame de Paris* is provided by WHU-TLS in the following link: http://3s.whu.edu.cn/ybs/en/benchmark.htm. The DURAARK project established a unique collection of datasets from the architectural domain, where *Room* is chosen from it. To verify the closure characteristic of the proposed algorithm, we use the Polyfit dataset (Nan and Wonka, 2017) to verify the closure problem. The dataset includes *Polyfit_4e*, *Bottom* and *Apartment*, etc.

Point cloud resolution, as one of the attributes of the point clouds, determines the ability to extract target details from the point clouds and is an important indicator of the accuracy of the reconstructed model. Here, the resolution is understood as the average distance between measurement points, according to the Point Cloud Library in the following link http://pointclouds.org/documentation/tutorials/correspondence_grouping.html. Table 1 shows the statistics of the processed scenes.

### 5.3. Input Parameter Settings

Since the data_set in the experiments are from different sources, the values of the parameters are entered manually according to the characteristics of each data. Specifically, the scale threshold ($s$) and the angle limit threshold ($al$) are required to segment the raw point cloud. If the distance between point pairs is less than $s$ and the angle is less than $al$, the two points are considered to come from the same plane. In the process of optimizing candidate sets, pairwise threshold ($pw$) represents the degree of association between primitives. According to different data sources, the values of $pw$ differ significantly, corresponding to the data characteristics. The value of the poplimit ($pl$) controls the size of the generated primitives and also controls the generation of small structures. If the number of points supporting the fitted face is less than a specific threshold $pl$, this face will not be generated. That is to say, that a face is generated only when the number of points contained in the fitted face is greater than $pl$. The number of iterations is represented by $it$, and $t$ is a number that is an exponential multiple of 2.

By analyzing the meaning of each parameter and measuring distance between point pairs, we found that the value of $s$ has a critical impact on the performance. In addition, the value of $pw$, $pl$ will affect the efficiency of each algorithm. Fig. 5 shows the comparison of experimental results obtained by different parameter settings, where the input parameters are shown in Table 2.

Table 2 presents the input parameters, which are expressed regarding each reconstruction result, shown in Fig. 5(a)–(h). The result is the number of faces in the model set reconstructed after the optimization. When the value of poplimit ($pl$) is increased, the number of generated faces decreases. The smaller the pairwise ($pw$), the more compact is the reconstructed model structure.

### 5.4. Baselines

The results of the proposed algorithm have been compared to mainstream algorithms, including RANSAC, discrete labeling based PEARL, constrained global structure GlobFit, RAPter-based method, and Polyfit (Nan and Wonka, 2017).

The traditional RANSAC method uses a probabilistic framework that is susceptible to outliers. The PEARL algorithm relies on the α-expansion library to reimplement the plane fitting. Compared to the two local reconstruction algorithms of RANSAC and PEARL, the GlobFit algorithm pays more attention to global relationships. The RAPter algorithm not only focuses on global relationships but also allows small key structures to remain and not be covered by the main structure. As for Polyfit, it focuses on capturing the entire closed structure by intersecting the initial primitive extensions.

### 5.5. Results and discussions

We evaluate the reconstruction results on point clouds with different sparsity, variable noise levels, and complexity. In general, our approach produced arrangements with higher regularity, and critically, guaranteeing a more closed polygon model. Fig. 6 shows the input point cloud, which is generated from the Smart 3D system (MVS), which is one of the mainstream commercial 3D reconstruction software based on multi-

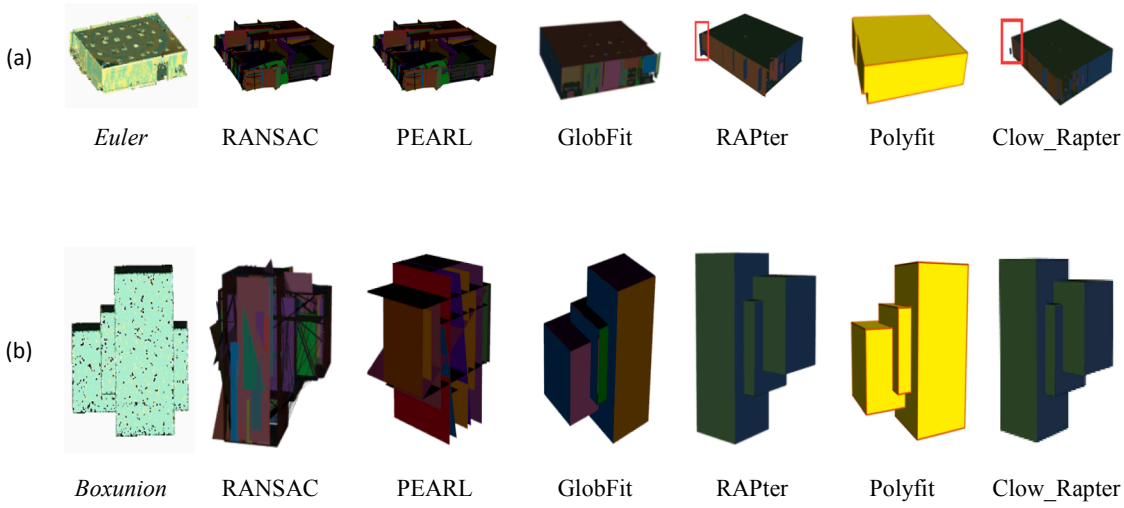



**Fig. 8.** Reconstruction results for *Euler* and *Boxunion* data sets. The *Euler* is relatively complex and contains many small structures, while the Boxuinon is more regular.

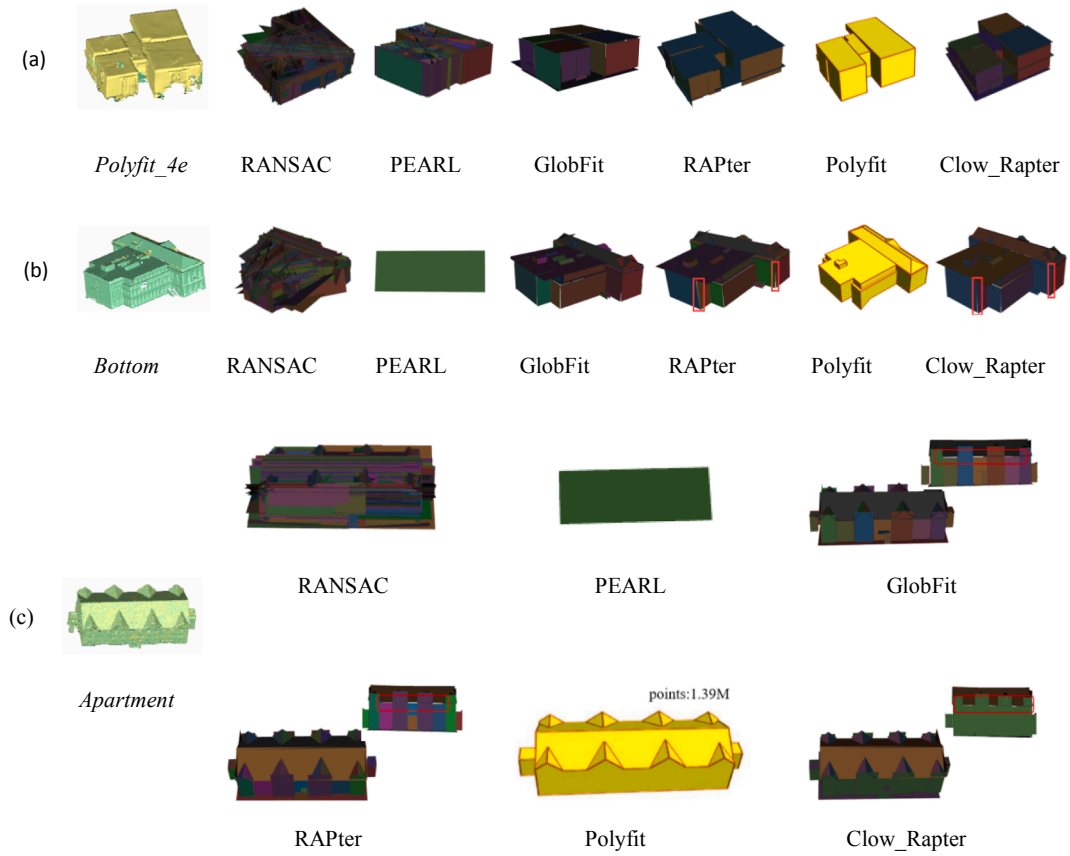



**Fig. 9.** Reconstruction results of the Polyfit benchmark data set.

view stereo system (MVS) (Seitz et al., 2006), using UAV (Unmanned Aerial Vehicle) aerial images. We assessed the method capability on non-uniformly sampled and sparsely sampled data and showed the impact of occlusion on the reconstruction results. We present for the proposed method the reconstruction consequence of data corrupted by outliers (see Fig. 7). Moreover, *Euler* is a good example of our approach to demonstrate the complexity and regularity of the building (see Fig. 8). We also conduct experiments on benchmark data_sets considering complex structures, as shown in Figs. 9 and 10. In Table 8, we also describe the quantitative results during model generation. Specifically, taking the number of generated faces as a reference object, our method is almost identical to the number of original planes, simultaneously generating a lightweight surface model. Meanwhile, we mainly consider the evaluation from three aspects: the coverage of inner points; the geometric fidelity; and the number of generated planes. The coverage means the ratio of the points assigned to the shape, and the geometric fidelity is the Root-Mean-Square-Error (RMSE) distance of the model to the inner points.

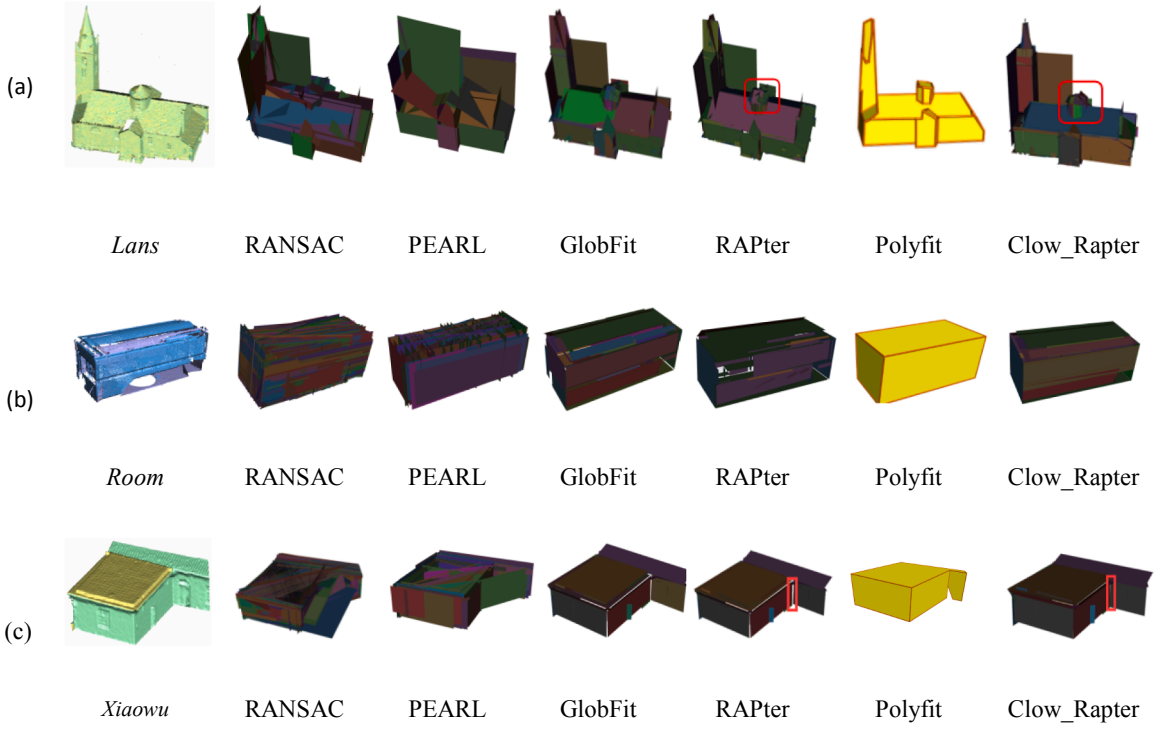



**Fig. 10.** The reconstruction results of LiDAR point clouds with complex structures.

Given the input point cloud S and the reconstructed model M, the coverage is defined through Eq. (9).

$$coverage = \frac{|\{s|\forall s \in S, s.t. \|s - M\| < T\}|}{|S|} \times 100\% \tag{9}$$

where $s$ is a point in $\boldsymbol{S}$. ||s-M|| is the Euclidean distance between s and M, and $|\boldsymbol{S}|$ is the number of points in $\boldsymbol{S}$. T is a preset threshold. In the experiments, we found that the threshold should be related to the density of the model. Based on empirical tests, we adopted the value of T equal to 0.6 times the size of the point resolution. The RMSE is defined by Eq. (10).

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}\sum_{\forall s \in S_i}\|s - P_i\|^2}{\sum_{i=1}^{n}|S_i|}} \tag{10}$$

where $P_i$ is the primitive plane of the reconstructed model M. Each plane $P_i$ is fitted from the associated set of points $S_i$. $||s\text{-}P_i||$ indicates the Euclidean distance between the point and the primitive plane. $|S_i|$ denotes the number of points in the point cloud block $S_i$.

The assessment metrics coverage and RMSE are trade-off indicators, since higher and lower coverage may respectively result in overfitting and underfitting. Therefore, we evaluate the experimental results based on the two metrics together with the number of generated planes.

Moreover, Fig. 12 shows the relationship graphs about the planes in the final model, to illustrate the effect of the closed constraints of the proposed algorithm on the reconstruction results.

Fig. 6 shows the point clouds corrupted by non-uniform sparse sampling. The traditional RANSAC method uses a probabilistic framework that is susceptible to outliers. The PEARL algorithm relies on the published $\alpha$-expansion library to reimplement the plane fit. Compared to the two local reconstruction algorithms of RANSAC and PEARL, the GlobFit algorithm pays more attention to the global relationship. The RAPter algorithm not only focuses on global relationships but also allows critical and small structures to remain without being covered by the overall structure. The main purpose of Polyfit is to obtain the closed structure through the intersection extension of the initial primitives. As shown in Fig. 6(a), *Container* is a relatively simple point cloud data.

Focusing on local fittings, as can be verified, RANSAC generates more local planes, which are cluttered. Referring to the priories, PEARL began to pay attention to the regular arrangement between the planes. The overall structure is generated in PEARL, but there is a large gap in the middle. GlobFit, which considers the global structure, extracts the overall surface structure of buildings reliably. Namely, the consideration of the global structure may have a positive effect on generating the rule model. Based on the GlobFit, RAPter also considers the relationship between planes from a global perspective and retains more detailed structures. There are no more small structures in this example, not providing significant different results in the visualization results for the two algorithms. The most obvious difference is that there are gaps between adjacent faces in the RAPter, and the closure of the model is inferior. To get a more watertight model, the most prominent feature of Polyfit is to intersect the initial primitives to obtain a more compact model. However, this method is outstanding in closed data, but not ideal in non-closed data. As shown in Fig. 6(a), part of the structure is not generated with Polyfit, and the overall structure is not accurately recovered. In this case, the proposed algorithm obtained a more watertight model. Similarly, the RANSAC of Fig. 6(b) generates a lot of fragments, and PEARL has a slanted plane. Polyfit results in considerable errors in the reconstruction of the structure of *Contain_unit*, which is a more complicated than Contain, and it presents a sparse point cloud. In Fig. 6(c), RANSAC and PEARL provided only local fitting results of the data, without overall structure. As expected, Polyfit generates a closed overall structure. However, this method loses some small main structures. The proposed algorithm can effectively obtain a more watertight model and allow the preservation of structures in non-dominant directions. In Fig. 6(d), our method, Clow_Rapter, effectively sews the gaps between the adjacent surfaces, as presented in the RAPter method.

In particular, *Bungalow* and Low building belong to low-rise buildings, but the reconstruction effect is quite different. Compared to *Bungalow*, Low building contains more junks and occlusions caused by poles in front of the buildings and railings on two-story balconies. The pole was involved in the reconstruction, which results in a collapsed

**Table 3**
Quantitative evaluation of surface reconstruction in aerial image point clouds.

| Method | *Container* | | | *Contain_unit* | | | *Tower* | | | *Bunglow* | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Cover (%) | RMSE (cm) | Prim | Cover (%) | RMSE (cm) | Prim | Cover (%) | RMSE (cm) | Prim | Cover (%) | RMSE (cm) | Prim |
| RANSAC | 99.89 | 5.2 | 483 | 89.87 | 11.1 | 650 | 85.31 | 8.3 | 522 | 63.51 | 24.8 | 581 |
| PEARL | 82.68 | 13.9 | 8 | 22.86 | 7.1 | 27 | 56.38 | 8.5 | 179 | 61.10 | 21.4 | 27 |
| GlobFit | 99.07 | 12.0 | 8 | 85.47 | 9.9 | 24 | 69.67 | 9.3 | 608 | 81.41 | 22.8 | 28 |
| RAPter | 98.93 | 12.3 | 8 | 82.41 | 10.1 | 18 | 76.14 | 8.3 | 638 | 78.50 | 23.1 | 20 |
| Polyfit | 70.24 | 7.6 | 6 | 50.14 | 7.3 | 9 | 60.48 | 8.5 | 27 | 55.21 | 26.4 | 16 |
| Clow_Rapter | 98.76 | 12.3 | 7 | 83.28 | 9.4 | 20 | 79.45 | 7.8 | 164 | 73.11 | 24.3 | 21 |

**Table 4**
Quantitative evaluation of surface reconstruction in LiDAR point clouds with noise.

| Method | *Empire* | | | *Boxunion_noise* | | |
|---|---|---|---|---|---|---|
| | Cover (%) | RMSE (cm) | Prim | Cover (%) | RMSE (cm) | Prim |
| RANSAC | 25.46 | 1.8 | 5 | 98.35 | 6.2 | 205 |
| PEARL | 94.40 | 2.7 | 62 | 34.26 | 5.7 | 13 |
| GlobFit | 64.76 | 1.5 | 13 | 93.75 | 7.2 | 123 |
| RAPter | 79.34 | 1.3 | 163 | 59.63 | 7.3 | 26 |
| Polyfit | 80.61 | 1.5 | 288 | 60.21 | 7.2 | 18 |
| Clow_Rapter | 79.94 | 1.1 | 324 | 59.39 | 7.2 | 25 |

structure. Consequently, the original structure of the building was destroyed. It can be seen that for sparsely sampled data, occlusion and junks have a significant impact on the reconstruction algorithm, and our algorithm cannot be completely robust.

To further evaluate the performance of the reconstruction results, Table 3 shows the quantitative comparison of each method. Note that RANSAC is capable of fitting regular data, such as *Container*, *Contain_unit*, and *Tower*. However, when the point cloud contains a high percentage of outlier or noise points, the performance of RANSAC is significantly decreased. As shown in Fig. 6(a)–(d), many redundant fragments are generated since the percentage of noise points is high. As shown in Table 3, RANSAC achieved the highest value of coverage because it mainly focuses on the data fitting. However, RANSAC generated 483 planes, which is different from the number of surfaces generated by the other methods. The results show that the reconstructed model may overfit if the fitting scheme only relies on data distribution. Moreover, the results of PEARL are unstable, while GlobFit achieved better results since it enables the algorithm to capture the overall structure of the building. Note that Polyfit cannot generate the original structure on non-closed data, and the contribution rate of the interior-points is low. By constraining the structure information, RAPter guarantees the coverage of the interior-points ensuring the global structure. The proposed Clow_Rapter can get a relatively large interior-point support rate while maintaining a smaller RMSE. In other words, the proposed Clow_Rapter gives a good trade-off between fitting error and prior shape.

For LiDAR point clouds corrupted by outliers on *Empire* and *Boxunion_noise* (Fig. 7), we verified the effective reconstruction performance of the proposed algorithm. Our algorithm is capable of recovering the global structure from data with different levels of noise. And more importantly, the resulting model structure is more compact. For *Empire*, most notably, the top of the building is a cylinder (Fig. 7(a)), approximated by planes that support the global structure. RANSAC oversimplifies the geometry in the process of local fitting due to a large amount of noise. However, as shown in *Boxunion_noise* (Fig. 7(b)), it generates more local messy structures, but small structures and inclined planes were not generated. Therefore, RANSAC has unstable reconstruction results for data with a lot of noise. PEARL generates more regular arrangements for planes. However, due to the balance of complexity and smoothness considered in the energy function, some of the main structures of the building are missing, as shown in Fig. 7(a) and (b) for PEARL. In contrast, GlobFit has better performance in the structural integrity of the model. Unfortunately, it cannot handle a lot of outliers. Consequently, more small details are missing in the top of the building. As shown in Fig. 7(a) (GlobFit), RAPter is closer to the original structure of the building. However, the top structure is dispersed, resulting in a poor closure of the model. Clow_Rapter, proposed in this paper, can effectively generate a more compact and complete structure of the building (see Clow_Rapter in Fig. 7(a)). Polyfit guarantees closure, but it also lacks some non-dominant structure. The data of *Boxunion_noise* also restores the original structure better than Clow_Rapter (Fig. 7(b)). Polyfit has almost perfect performance on high-precision data for simple structures with rules than other all methods.

Table 4 presents the quantitative evaluation of the surface reconstruction in LiDAR point clouds with a high percentage of noise points. The results of RANSAC and PEARL are unstable since they only focus on local plane fitting. For example, the coverage of the RANSAC in *Empire* is only 25.26%, while the score is 98.35% in *Boxunion_noise*. In *Empire* data, with a complex structure, RANSAC generates very few faces, while in a simple *Boxunion_noise*, it generates a lot of fragments. On the other hand, RAPter, Polyfit, GlobFit and Clow_Rapter are more balanced for different data, considering the coverage and the RMSE. The main reason is that both point clouds are closed structures. As a result, Polyfit is comparable to RAPter and Clow_Rapter. In *Empire* with complex structure, Clow_Rapter generated 324 faces, which can retain more details, as shown in Table 4. However, in the simple structure of *Boxunion_noise*, it can generate a lightweight model. In general, the proposed Clow_Rapter algorithm can reconstruct the global structure from noisy data with closure constraints.

The following experiment aims to verify the result of the proposed algorithm in point clouds with a more regular structure. For point clouds with regular structures, the *Euler* (Fig. 8(a)) is relatively complex and contains many small structures, while the Boxuinon is more regular. For rule point clouds, RANSAC and PEARL still generate a lot of fragments and do not restore the whole structure. GlobFit shows the best performance to reconstruct the objects due to maintaining global relationships. In the *Euler* with complex structures, Globfit not only reconstructs the complete structure more accurately but also retains more fine structures. However, RAPter and Clow_Rapter failed to generate those fragments despite the restoration of the whole structure of the building, which may be related to the number of iterations that generated the

**Table 5**
Quantitative evaluation of surface reconstruction in LiDAR point clouds with a regular structure.

| Method | *Euler* | | | *Boxunion* | | |
|---|---|---|---|---|---|---|
| | Cover (%) | RMSE (cm) | Prim | Cover (%) | RMSE (cm) | Prim |
| RANSAC | 89.17 | 2.2 | 301 | 97.86 | 3.0 | 672 |
| PEARL | 99.40 | 1.4 | 236 | 52.67 | 4.2 | 19 |
| GlobFit | 99.98 | 1.0 | 315 | 100 | 0.9 | 20 |
| RAPter | 89.37 | 1.4 | 151 | 100 | 0.9 | 20 |
| Polyfit | 83.13 | 1.5 | 55 | 99.46 | 1.1 | 20 |
| Clow_Rapter | 95.31 | 1.2 | 236 | 100 | 0.9 | 20 |

**Table 6**
Quantitative evaluation of the reconstruction results of the Polyfit dataset.

| Method | *Polyfit_4e* | | | *Bottom* | | | *Apartment* | | |
|---|---|---|---|---|---|---|---|---|---|
| | Cover (%) | RMSE (cm) | Prim | Cover (%) | RMSE (cm) | Prim | Cover (%) | RMSE (cm) | Prim |
| RANSAC | 97.67 | 8.7 | 871 | 99.88 | 10.6 | 1304 | 99.99 | 15.9 | 935 |
| PEARL | 52.89 | 10.8 | 67 | – | | 1 | – | | 1 |
| GlobFit | 71.98 | 11.2 | 68 | 97.79 | 14.3 | 237 | 95.61 | 20.7 | 478 |
| RAPter | 71.08 | 11.5 | 62 | 94.56 | 15.5 | 142 | 95.59 | 21.1 | 55 |
| Polyfit | 80.38 | 11.6 | 8 | 69.72 | 16.6 | 64 | 56.04 | 17.0 | 7 |
| Clow_Rapter | 69.56 | 11.2 | 63 | 91.71 | 18.8 | 132 | 82.82 | 25.4 | 55 |

model. Clow_Rapter optimizes the intersection between planes based on RAPter. Polyfit generated a rough model framework as expected. Besides, for a simple point cloud of the rule structure, GlobFit, RAPter, Polyfit, and Clow_Rapter in Fig. 8(b) gives a perfect reconstruction result considering the global structure between the primitives.

Experimental results indicate that Clow_Rapter is robust to favor globally consistent closed regular arrangement planes of the reconstructed object as much as possible; even the residuals of the primitive fit become larger. In contrast, the plane adjacent to the final model is more compact in the data corrupted by non-uniform sampling, noise, and complex structures. As for buildings with regular structure, all methods achieve a high interior-point coverage rate of more than 90%, as shown in Table 5. Note that our method achieves the smallest geometric error, and even can achieve perfect reconstruction, such as in Boxuninon. In the *Euler* data with complex structure, the proposed method generates relatively more planes and can retain more details of the surface model.

We also test all methods on the Polyfit benchmark dataset, which comes from the Polyfit project. The first experiment is based on *Polyfit_4e*, which is formed by the combination of several close structures. As shown in Fig. 9(a), due to the influence of the acquisition technology or noise, some points are missing in the original point cloud. From the experimental results, we can see that RANSAC and PERAL present overfitting, as too many redundant planes are generated to achieve a high coverage. On the other hand, although the results of RAPter and GlobFit seem promising, they did not provide a closed 3D model. Noteworthy that closed-based methods such as Polyfit and the proposed Clow-Rapter acquired a closed surface model. That being said, Polyfit has problems of underfitting, since the number of the plane is not enough to cover the surface points. The same conclusion can be drawn in the data of *Bottom*. One exception is that PEARL completely failed in this case. In the case of *Apartment* (Fig. 9(c)), there are complex planes, which have different density and coverage, in the point cloud. Note that RANSAC and PERAL are totally collapsed. GlobFit and RAPter present overfitting, while Polyfit presents underfitting. The proposed close constraint effectively reduces the gap at the side of adjacent parts.

Table 6 presents the quantitative evaluation of the reconstruction results of the Polyfit dataset. The typical case is that RANSAC has strong capabilities in data fitting, but it has difficulties in recovering complex data since it ignores the prior of the global structure. Note that the proposed algorithm can improve the geometric fidelity when the contribution rate of the interior points to the structure is reduced, and has an obvious effect on the global structure, as shown in Fig. 9 and Table 6.

In the following experiments, we focus on evaluating the performance of each method based on point clouds with complex structures. As shown in the first column of Fig. 10, *Lans* has tapered structures and octagonal pyramids, and *Room* includes not only surface points, but also an indoor scene, which may interfere in the surface reconstruction. As for *Xiaowu*, the two houses in the scene are joined together. There is no doubt that the surface reconstruction of these data has great challenges. In the second and the third column of Fig. 10, RANSAC and PEARL generate redundant fragments and cannot obtain the global structure. As shown in Fig. 10 (a), GlobFit and RAPter used global structure and then has restored the overall structure of the building, while the output model is not closed. Clow_Rapter can completely reconstruct the building, where the generated anise stars and *Tower* tops in the data of *Lans* are better compared to other methods. It is worth noting that the octagon of Clow_Rapter is more standardized and compact. At the corner of *Xiaowu*, our method improves the closure of the joint, as shown in Fig.10 (c). Note that Polyfit failed in the data of *Xiaowu*, which is a non-closed point cloud. In general, Polyfit is not applicable to data with complex data structures.

Table 7 represents the quantitative evaluation of the reconstruction performance. Overall, our method maintains the smallest geometric errors. Notably, in the data of *Lans*, RAPter and Clow_Rapter have similar interior-point coverage, and Clow_Rapter can get a more watertight surface model. In other words, our method guarantees the global structure of the model without losing the contribution rate of the interior points. It needs to be explained that *Room* is a point cloud data containing indoor scenes, such as tables and chairs. Polyfit only generates closed faces, and some small structures disappear. The *Xiaowu* data is a non-closed data, and the eaves structure is not generated in the model. Therefore, the point coverage of their model is low.

Objectively speaking, our method is not perfect, and the reconstruction performance on some other data is poor and even the original overall structure cannot be restored. *Nola* contains many small structures (Fig.11 (a)). *Notre-Dame de Paris* has a typical western architectural style and has a more complex structure, as shown in Fig. 11(b). The point cloud reconstruction of these data has great challenges. We use various methods to try to reconstruct these buildings. In particular, the reconstruction of *Notre-Dame de Paris* is difficult. Although our method cannot fully reproduce various shapes, it can clearly locate the overall structure layout, which can lay the foundation for the surface reconstruction in point clouds with complex structures.

Table 8 shows the number of faces in the model generation stage of each method in a different category of point clouds. The number of faces

**Table7**
Quantitative evaluation of surface reconstruction in point clouds with complex structure.

| Method | *Lans* | | | *Room* | | | *Xiaowu* | | |
|---|---|---|---|---|---|---|---|---|---|
| | Cover (%) | RMSE (cm) | Prim | Cover (%) | RMSE (cm) | Prim | Cover (%) | RMSE (cm) | Prim |
| RANSAC | 49.74 | 1.3 | 477 | 99.69 | 4.5 | 712 | 90.04 | 5.1 | 800 |
| PEARL | 31.54 | 1.0 | 523 | 60.85 | 7.2 | 65 | 30.58 | 4.8 | 243 |
| GlobFit | 90.81 | 1.5 | 300 | 84.91 | 5.9 | 76 | 77.57 | 6.9 | 214 |
| RAPter | 96.74 | 1.4 | 544 | 69.17 | 6.9 | 57 | 75.74 | 6.9 | 229 |
| Polyfit | 91.98 | 1.5 | 51 | 15.63 | 6.3 | 22 | 21.59 | 4.6 | 28 |
| Clow_Rapter | 94.23 | 1.0 | 246 | 83.15 | 6.1 | 59 | 73.52 | 2.4 | 230 |

**Table 8**
Comparison of experimental results: the numbers of input planar segments (initialization), primitives that generated the candidate set (candidate), and the regular set (optimization).

| Category | Name (points) | Method | Initialization | Candidate | Optimization |
|---|---|---|---|---|---|
| Image-Based Point Cloud | *Container* (81,779) | RANSAC | 8 | – | 483 |
| | | Pearl | 8 | – | 8 |
| | | Globfit | 8 | – | 8 |
| | | RAPter | 8 | 16 | 7 |
| | | clow_Rapter | **8** | 24 | **7** |
| | | Polyfit | 8 | 587 | 13 |
| LiDAR points cloud | *Boxunion* (100,000) | RANSAC | 20 | – | 672 |
| | | Pearl | 20 | – | 19 |
| | | Globfit | 20 | – | 20 |
| | | RAPter | 20 | 142 | 21 |
| | | clow_Rapter | **20** | 136 | **20** |
| | | Polyfit | 20 | 882 | 20 |
| LiDAR points cloud with noise | *Empire* (1,200,000) | RANSAC | 327 | – | 5 |
| | | Pearl | 327 | – | 62 |
| | | Globfit | 327 | – | 13 |
| | | RAPter | 327 | 2966 | 326 |
| | | clow_Rapter | **327** | 2908 | **320** |
| | | Polyfit | 327 | 5217 | 43 |

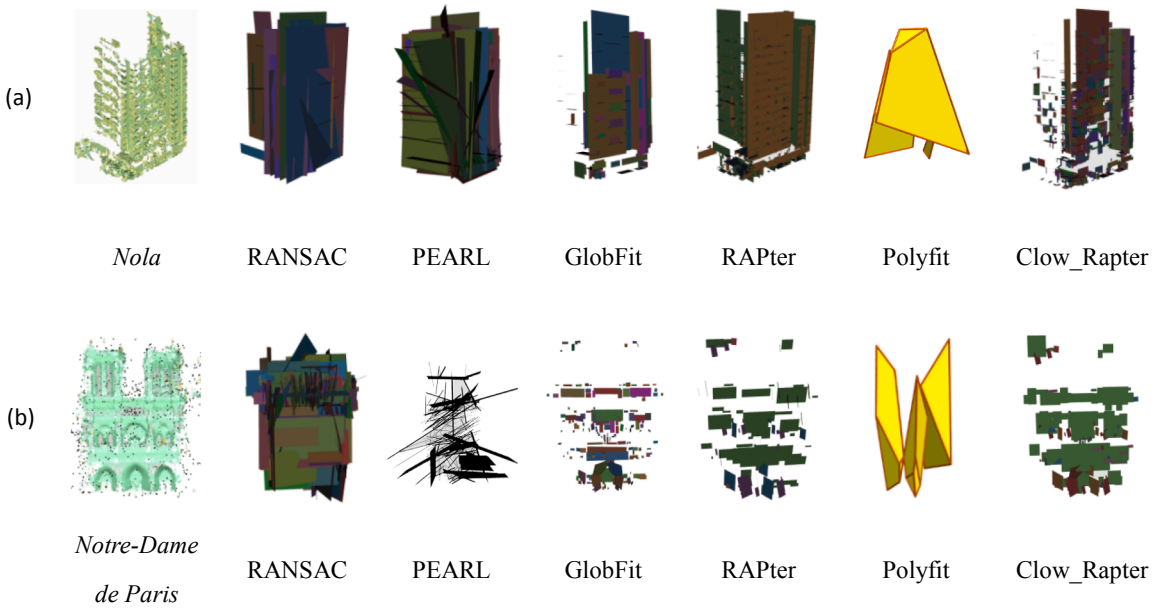


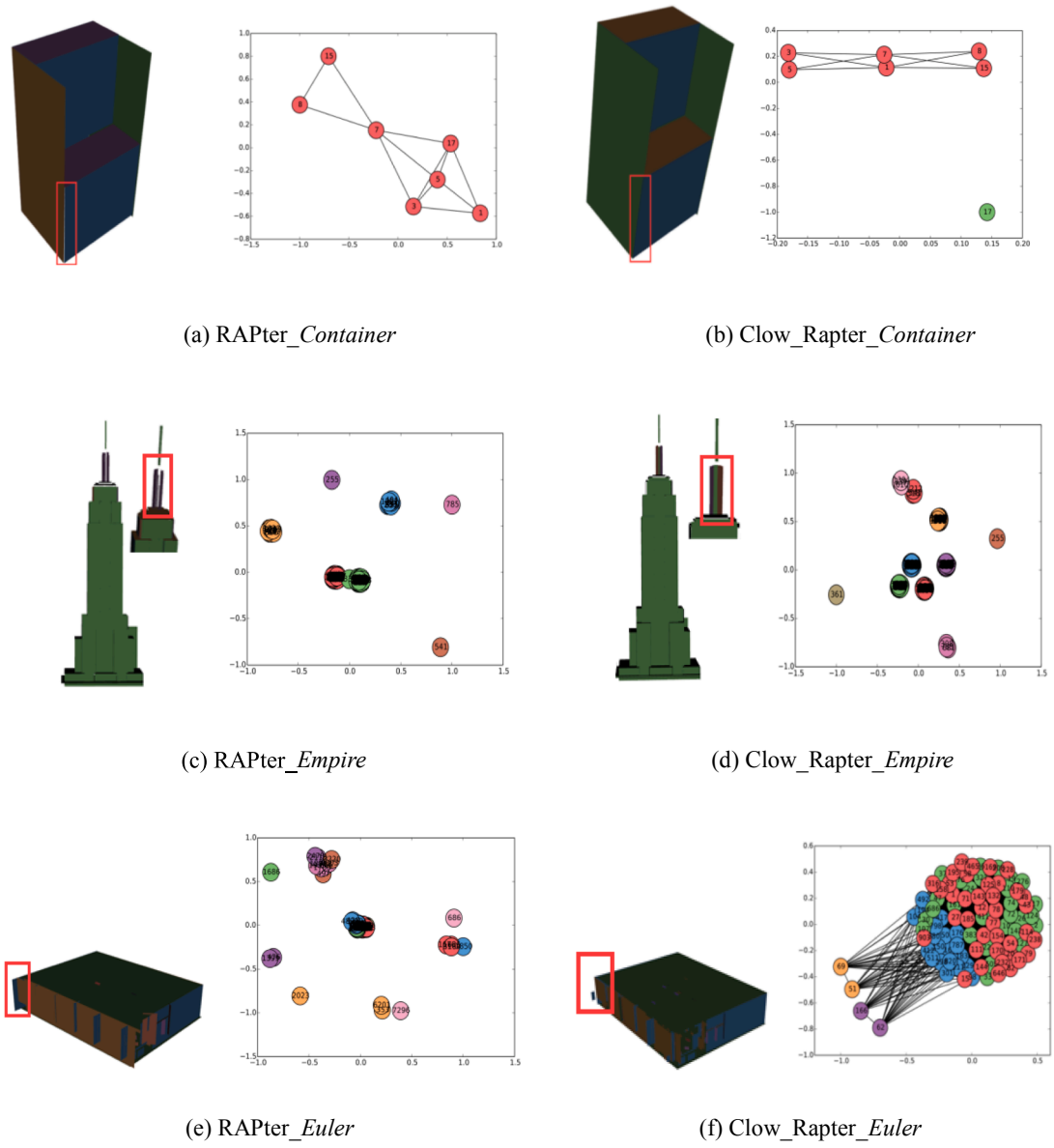

**Fig. 11.** The reconstruction results of other LiDAR point clouds.

optimized and selected based on the candidate model set should ideally be similar to the initial set of that.

The results show that the number of models generated by the RANSAC algorithm differs significantly from the initial set of fits. For example, for *Container*, the number of final model sets for *Container* is more than six times the initial fit due to the generation of many fragments, as shown in Fig. 6(a). Since the excessive presence of outliers in *Empire* (Fig. 7(a)), RANSAC over-simplified the geometry. PEARL discovered the main directions, which is close to the number of input planar segments, but the planes returned by PEARL, are not perfectly aligned. Given more, it cannot effectively process data containing a large number of outliers. As a result, the tiny main structure on top of the building was lost (Fig. 7(a)).

In contrast, RAPter can effectively generate the main surface from the sparse point cloud and the data disturbed with noise, which shows its capacity to deal with large scale scenes. The most prominent is that it can identify and retain important and non-dominant structures in the main structural context. The main disadvantages are gaps between adjacent faces resulting in poor overall closure of the model.

For Polyfit, more candidate sets are generated by the intersection of adjacent faces, which is even more than 50% off the candidate set of other algorithms. And the regular set is far from the initialization. As shown in Figs. 6 and 7, the regular set loses many smaller details in the *Container* data and the *Empire* data. Considering the closed constraint, the number of candidate set generated by Clow_Rapter (320) is less than that of RAPter (326). In general, both algorithms are consistent with the number of input planar segments. Still, the experimental results of the visualization show that the Polyfit reconstruction effect is better in constraining the overall model closure. However, the Clow_Rapter performs better in terms of balancing the global structure and closure. To more intuitively show the effect of the closed constraint on the relationship between primitives, the graphical representation is shown in Fig. 12. A circle indicates a plane, and a line between dots indicates the relationship between planes. It is worth noting that different colored dots are represented by different initial primitives through rigid-body transformation. In other words, this means that the more colored dots, the more initial primitives retained infinal model sets after selection from the candidate set, which causes more initial primitives and



(a) RAPter_*Container* (b) Clow_Rapter_*Container*

(c) RAPter_*Empire* (d) Clow_Rapter_*Empire*

(e) RAPter_*Euler* (f) Clow_Rapter_*Euler*

**Fig. 12.** Relationship between the primitives in the reconstruction results of the RAPter and Clow_Rapter algorithms. In Figs. (a)–(f), the left side is the reconstruction model of the algorithm, and the right side is a graphical representation of the relationship between the corresponding planes.

structures are preserved as well as the resulting model structure is richer and more complete.

In Fig. 6, RAPter generates seven faces in *Container*, all of which are obtained by transforming one initial primitive. As shown in Fig. 12, faces are marked by two colors, red and green, respectively, indicating that the whole model consists of two initial primitive transformations. As we can see, there is a gap when considering the RAPter method in Fig. 12(a), representing that the fitting at this position by the same primitive is not perfect. It is more suitable by another primitive, such as in Fig. 12(b). Similarly, the crack at the top of the *Empire* disappears in Fig. 12(d), and the structure is more compact than (c). From the relationship diagram, the points in Fig. 12(d) (*Empire*) have more types of color than (c), retaining more initial structure. So the top structure of the *Empire* is more regular, which improves the closure of the overall model. As for the *Euler* with complex structures, RAPter has more colors than Clow_Rapter, so RAPter retains more fine structures than Clow_Rapter, such as the fragments of the roof. Furthermore, the point distance in RAPter is close. In contrast, the distribution of the Clow_Rapter is more uniform in Fig. 12(f).

## 6. Conclusions

In this paper, we introduced a novel building reconstruction algorithm based on closed constraints to obtain watertight and complete building surface model, named Clow_Rapter. The algorithm considers the closed constraints of polygon boundaries to perform surface reconstruction based on regular arrangements of planes. The pipeline of the proposed algorithm includes candidate planes generation from an input point cloud, candidate planes regulation, closure constrain, and surface model optimization. The main contribution is to detect the adjacency of intersecting faces, then to enforce the boundary closure of the primitive in the process surface model generation. The experimental results show that our algorithm improves the closeness of the global structure without sacrificing the coverage rate of interior points and geometric errors. Significantly, the watertight structure of the building is nearly guaranteed in the point clouds corrupted by non-uniform, sparse sampling, or outliers. Besides, Clow_Rapter supports lightweight model generation, which can identify and retain the structure that is consistent with a priori geometric structures. That being said, our algorithm may

be over-regularized in the situation of building with very complex surfaces. Therefore, our future work will focus on the analysis of the trade-off between closure regulation and data fitting error.

## 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.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China under grants of 41971424, 61702251, 61701191 and U1605254, the Key Technical Project of Fujian Province under Grant No. 2017H6015, the Natural Sciences and Engineering Research Council of Canada under a grant of 50503-10284, the Science and Technology Project of Xiamen under Grant No. 3502Z20183032, and the Key Project of Xiamen Southern Oceanographic Center under grants No. 18CZB033HJ11.

## References

Achlioptas, P., Fan, J., Hawkins, R. X., Goodman, N. D., Guibas, L.J., 2019. ShapeGlot: Learning Language for Shape Differentiation. arXiv preprint arXiv:1905.02925.
Alexa, M., Behr, J., Cohen-Or, D., Fleishman, S., Levin, D., Silva, C.T., 2003. Computing and rendering point set surfaces. IEEE Trans. Vis. Comput. Graph. 9 (1), 3–15. https://doi.org/10.1109/TVCG.2003.1175093.
Alliez, P., Cohen-Steiner, D., Tong, Y., Desbrun, M., 2007. Voronoi-based variational reconstruction of unoriented point sets. In: Proceedings of the 5th Eurographics Symposium on Geometry Processing, pp. 39–48.
Amenta, N., Bern, M., Kamvysselis, M., 1998. A new Voronoi-based surface reconstruction algorithm. In: Proceedings of the 25th annual conference on Computer graphics and interactive techniques, pp. 415–421.
Arikan, M., Schwärzler, M., Flöry, S., Wimmer, M., Maierhofer, S., 2013. O-Snap: Optimization-based snapping for modeling architecture. ACM Trans. Graph. 32, 6. https://doi.org/10.1145/2421636.2421642.
Arsalan Soltani, A., Huang, H., Wu, J., Kulkarni, T.D., Tenen-baum, J.B., 2017. Synthesizing 3D shapes via modeling multiview depth maps and silhouettes with deep generative networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1511–1519.
Berger, M., Tagliasacchi, A., Seversky, L.M., Alliez, P., Guen-nebaud, G., Levine, J.A., Sharf, A., Silva, C.T., 2017. A survey of surface reconstruction from point clouds. Comput. Graph. Forum. 36, 301–329.
Bruno, F., Bruno, S., De Sensi, G., Luchi, M.-L., Mancuso, S., Muzzupappa, M., 2010. From 3D reconstruction to virtual reality: A complete methodology for digital archaeological exhibition. Journal of Cultural Heritage. 11, 42–49. https://doi.org/10.1016/j.culher.2009.02.006.
Chen, D., Wang, R., Peethambaran, J., 2017. Topologically aware building rooftop reconstruction from airborne laser scanning point clouds. IEEE Trans. Geosci. Remote Sens. 55, 7032–7052. https://doi.org/10.1109/tgrs.2017.2738439.
Chen, J., Chen, B., 2008. Architectural modeling from sparsely scanned range data. Int. J. Comput. Vision 78, 223–236. https://doi.org/10.1007/s11263-007-0105-5.
Chum, O., Matas, J., 2005. Matching with prosac - progressive sample consensus. In: 2005 IEEE Conference on Computer Vision and Pattern Recognition, pp. 220–226.
Diebel, J.R., Thrun, S., Brünig, M., 2006. A Bayesian method for probable surface reconstruction and decimation. ACM Trans. Graph. 25, 39–59. https://doi.org/10.1145/1122501.1122504.
Fan, H., Su, H., Guibas, L.J., 2017. A point set generation network for 3D object reconstruction from a single image. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 605–613.
Furukawa, Y., Curless, B., Seitz, S.M., Szeliski, R., 2009. Reconstructing building interiors from images. In: 2009 International Conference on Computer Vision, pp. 80–87. https://doi.org/10.1109/ICCV.2009.5459145.
Gal, R., Shamir, A., Hassner, T., Pauly, M., Cohen-Or, D., 2007. Surface reconstruction using local shape priors. In: Proceedings of the 5th Eurographics Symposium on Geometry Processing, pp. 253–262.
Ganapathi-Subramanian, V., Diamanti, O., Pirk, S., Tang, C., Niessner, M., Guibas, L., 2018. Parsing geometry using structure-aware shape templates. In: 2018 International Conference on 3D Vision, pp. 672–681.
Groueix, T., Fisher, M., Kim, V.G., Russell, B.C., Aubry, M., 2018. A papier-mâché approach to learning 3D surface generation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 216–224.
Guerrero, P., Kleiman, Y., Ovsjanikov, M., Mitra, N.J., 2018. PCPNET learning local shape properties from raw point clouds. Comput. Graph. Forum. 37, 75–85.
Hartley, R., Zisserman, A., 2003. Multiple view geometry in computer vision. Cambridge University Press.
Hoppe, H., DeRose, T., Duchamp, T., Halstead, M., Jin, H., McDonald, J., Schweitzer, J., Stuetzle, W., 1994. Piecewise smooth surface reconstruction. In: Proceedings of the 21st annual conference on Computer graphics and interactive techniques, pp. 295–302.
Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., Stuetzle, W., 1992. Surface reconstruction from unorganized points. ACM. SIGGRAPH Comput. Graph. 26 (2), 71–78.
Isack, H., Boykov, Y., 2012. Energy-based geometric multi-model fitting. Int. J. Comput. Vision 97, 123–147. https://doi.org/10.1007/s11263-011-0474-7.
Jarzkabek-Rychard, M., Borkowski, A., 2016. 3D building reconstruction from ALS data using unambiguous decomposition into elementary structures. ISPRS J. Photogramm. Remote Sens. 118, 1–12.
Jenke, P., Krückeberg, B., Straßer, W., 2008. Surface reconstruction from fitted shape primitives. In: Vision, Modeling, & Visualization Conference, pp. 31–40.
Jenke, P., Wand, M., Bokeloh, M., Schilling, A., Straßer, W., 2006. Bayesian point cloud reconstruction. Comput. Graph. Forum 25, 379–388.
Kalojanov, J., Lim, I., Mitra, N., Kobbelt, L., 2019. String-based synthesis of structured shapes. Comput. Graph. Forum 38, 27–36.
Kazhdan, M., Bolitho, M., Hoppe, H., 2006. Poisson surface reconstruction. In: Proceedings of the 4th Eurographics Symposium on Geometry Processing, pp. 61–70.
Lafarge, F., Alliez, P., 2013. Surface reconstruction through point set structuring. Comput. Graph. Forum 32, 225–234.
Li, B., Schnabel, R., Jin, S., Klein, R., 2009. Variational surface approximation and model selection. Comput. Graph. Forum 28, 1985–1994.
Li, C.-L., Zaheer, M., Zhang, Y., Poczos, B., Salakhutdinov, R., 2018. Point cloud gan. arXiv preprint arXiv:1810.05795.
Li, H., Li, Y., Yu, R., Sun, J., Kim, J., 2018. Surface reconstruction from unorganized points with $l_0$ gradient minimization. Computer Vis. Image Unders. 169, 108–118.
Li, J., Xu, K., Chaudhuri, S., Yumer, E., Zhang, H., Guibas, L., 2017. Grass: Generative recursive autoencoders for shape structures. ACM Trans. Graph. 36 (4) https://doi.org/10.1145/3072959.3073613.
Li, M., Rottensteiner, F., Heipke, C., 2019. Modelling of buildings from aerial LiDAR point clouds using TINs and label maps. ISPRS J. Photogramm. Remote Sens. 154, 127–138. https://doi.org/10.1016/j.isprsjprs.2019.06.003.
Li, Y., Vinyals, O., Dyer, C., Pascanu, R., Battaglia, P., 2018. Learning deep generative models of graphs. arXiv preprint arXiv:1803.03324.
Li, Y., Wu, X., Chrysathou, Y., Sharf, A., Cohen-Or, D., Mitra, N.J., 2011. Globfit: Consistently fitting primitives by discovering global relations. In: International Conference on Computer Graphics and Interactive Techniques. https://doi.org/10.1145/1964921.1964947, pp. 30(4).
Lin, H., Gao, J., Zhou, Y., Lu, G., Ye, M., Zhang, C., Liu, L., Yang, R., 2013. Semantic decomposition and reconstruction of residential scenes from LiDAR data. ACM Trans. Graph. 32, 66. https://doi.org/10.1145/2461912.2461969.
Maćkiewicz, A., Ratajczak, W., 1993. Principal components analysis (PCA). Comput. Geosci. 19, 303–342. https://doi.org/10.1016/0098-3004(93)90090-R.
Mo, K., Guerrero, P., Yi, L., Su, H., Wonka, P., Mitra, N., Guibas, L.J., 2019. StructureNet: Hierarchical graph networks for 3D shape generation. arXiv preprint arXiv: 1908.00575.
Mo, K., Zhu, S., Chang, A.X., Yi, L., Tripathi, S., Guibas, L.J., Su, H., 2019b. PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 909–918.
Monszpart, A., Mellado, N., Brostow, G.J., Mitra, N.J., 2015. RAPter: rebuilding man-made scenes with regular arrangements of planes. ACM Trans. Graph. 34 (4), 103: 1–103:12. https://doi.org/10.1145/2766995.
Musialski, P., Wimmer, M., Wonka, P., 2012. Interactive coherence-based facade modeling. Comput. Graph. Forum. 31, 661–670.
Nan, L., Jiang, C., Ghanem, B., Wonka, P., 2015. Template assembly for detailed urban reconstruction. Comput. Graph. Forum. 34, 217–228.
Nan, L., Sharf, A., Zhang, H., Cohen-Or, D., Chen, B., 2010. SmartBoxes for interactive urban reconstruction. ACM Trans. Graphics. https://doi.org/10.1145/1778765.1778830.
Nan, L., Wonka, P., 2017. PolyFit: Polygonal surface reconstruction from point clouds. In: 2017 International Conference on Computer Vision, pp. 2353–2361.
Ni, K., Jin, H., Dellaert, F., 2009. Groupsac: Efficient consensus in the presence of groupings. In: 2009 International Conference on Computer Vision, pp. 2193–2200.
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. https://doi.org/10.1016/j.isprsjprs.2019.03.017.
Oesau, S., Lafarge, F., Alliez, P., 2014. Indoor scene reconstruction using feature sensitive primitive extraction and graph-cut. ISPRS J. Photogram. Remote Sens. 90, 68–82. https://doi.org/10.1016/j.isprsjprs.2014.02.004.
Ohtake, Y., Belyaev, A., Seidel, H.-P., 2003. A multi-scale approach to 3D scattered data interpolation with compactly supported basis functions. In: 2003 Shape Modeling International, pp. 153–161.
Pauly, M., Mitra, N.J., Wallner, J., Pottmann, H., Guibas, L.J., 2008. Discovering structural regularity in 3D geometry. ACM Trans. Graph. 27 (3) https://doi.org/10.1145/1360612.1360642.
Rajput, A., Funk, E., Boerner, A., Hellwich, O., 2018. A regularized volumetric fusion framework for large-scale 3D reconstruction. ISPRS J. Photogramm. Remote Sens. 141, 124–136. https://doi.org/10.1016/j.isprsjprs.2018.04.020.
Schnabel, R., Wahl, R., Klein, R., 2007. Efficient ransac for pointcloud shape detection. Comput. Graph. Forum. 26, 214–226.
Seitz, S.M., Curless, B., Diebel, J., Scharstein, D., Szeliski, R., 2006. A comparison and evaluation of multi-view stereo reconstruction algorithms. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 519–528.
Tachella, J., Altmann, Y., Mellado, N., McCarthy, A., Tobin, R., Buller, G.S., Tourneret, J.-Y., McLaughlin, S., 2019. Real-time 3D reconstruction from single-

photon lidar data using plug-and-play point cloud denoisers. Nature Commun. 10 (1), 1–6. https://doi.org/10.1038/s41467-019-12943-7.

Verma, V., Kumar, R., Hsu, S., 2006. 3D building detection and modeling from aerial lidar data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2213–2220.

Wan, G., Sharf, A., 2012. Grammar-based 3D facade segmentation and reconstruction. Comput. Graphics 36 (4), 216–223. https://doi.org/10.1016/j.cag.2012.02.003.

Wang, J., Fang, T., Su, Q., Zhu, S., Liu, J., Cai, S., Tai, C.-L., Quan, L., 2015. Image-based building regularization using structural linear features. IEEE Trans. Vis. Comput. Graph. 22, 1760–1772. https://doi.org/10.1109/TVCG.2015.2461163.

Wong, H.S., Chin, T.-J., Yu, J., Suter, D., 2011. Dynamic and hierarchical multi-structure geometric model fitting. In: 2011 International Conference on Computer Vision, pp. 1044–1051.

Wu, B., Yu, B., Wu, Q., Yao, S., Zhao, F., Mao, W., Wu, J., 2017. A graph-based approach for 3D building model reconstruction from airborne LiDAR point clouds. Remote Sens. 9 (1), 92. https://doi.org/10.3390/rs9010092.

Xia, S., Chen, D., Wang, R., Li, J., Zhang, X., 2020. Geometric primitives in LiDAR point clouds: A review. IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. 13, 685–707. https://doi.org/10.1109/JSTARS.2020.2969119.

Yang, B., Lee, J., 2017. Improving accuracy of automated 3-D building models for smart cities. Int. J. Digit. Earth 12, 209–227. https://doi.org/10.1080/17538947.2017.1395089.

Yi, C., Zhang, Y., Wu, Q., Xu, Y., Remil, O., Wei, M., Wang, J., 2017. Urban building reconstruction from raw LiDAR point data. Computer-Aided Des. 93, 1–14. https://doi.org/10.1016/j.cad.2017.07.005.

Yumer, M.E., Kara, L.B., 2012. Co-abstraction of shape collections. ACM Trans. Graph. 31 (6), 166. https://doi.org/10.1145/2366145.2366185.

Zheng, Q., Sharf, A., Wan, G., Li, Y., Mitra, N.J., Cohen-Or, D., Chen, B., 2010. Non-local scan consolidation for 3D urban scenes. ACM Trans. Graph. 29 (4), 94–101.