Building Instance Mapping From ALS Point Clouds Aided by Polygonal Maps

Shaobo Xia, Sheng Xu, Ruisheng Wang, *Senior Member, IEEE*, Jonathan Li, *Senior Member, IEEE*, and Guanghui Wang, *Senior Member, IEEE*

***Abstract*— Building region extraction from ALS point clouds has been widely studied, whereas instance-level building mapping has been overlooked and remains unsolved. In this study, we present a method to extract individual buildings from ALS point clouds with the help of widely accessible polygonal footprints. The key idea is to merge roof segments to a set of building candidates, from which correct instances are selected by finding optimal matches between polygonal footprints and building candidates. The method has three steps: roof segmentation, building candidate generation, and instance-polygon matching. The method is tested on two large-scale scenes of different building types and can generally achieve high instance-level building mapping accuracy (around 90%) when there are large positioning errors (6.0 m) among polygons. Future work will focus on classification errors in preprocessing, shape inconsistency between point clouds and polygons, and building footprint delineation and updating in postprocessing.**

***Index Terms*— Building footprints, instance-level segmentation, OpenStreetMap (OSM), point clouds, polygon matching.**

I. Introduction

SIGNIFICANT progress has been made in building extraction from remote sensing data, such as from aerial images [1], [2] and airborne laser scanning (ALS) [3], [4]. Instance-level object detection or segmentation, which means that segmented regions that have the same class labels should be further separated into individual objects [5], is becoming increasingly important in 2-D images. This also applies to building instance mapping, which is the extraction of independent buildings from remote sensing data. Instance-level building extraction is the basis of much research and many applications. For instance, the number of buildings was counted at instance level [6]. Kang *et al.* [7] proposed a building type classification method for instance-level footprints. In building footprint delineation [8], [9] or 3-D model reconstruction [10], [11], input data should be point clouds of one building object. Instance-level building objects are widely used in change detection [12], building density analysis [13], building clustering [14], and scene recognition [15].

As a positive remote sensing technology, ALS has been widely used in building mapping. Previous studies [3], [16], [17] have focused on building region extraction from ALS point clouds, while instance-level building mapping has received little attention. Nevertheless, there exist some related studies that can be roughly grouped into rule-based and rectangle-assumption-based methods.

In the first group, Cote and Saeedi [18] segmented aerial images into homogeneous regions based on color information and identified building instances according to predefined rules, such as by roof sizes and shapes. Yu *et al.* [13] proposed an individual building segmentation method for an ALS-derived normalized digital surface model (nDSM), relying mainly on the spatial distance between buildings after removing ground and vegetation. Adjacent building instances are merged into one instance. Wang *et al.* [19] proposed building instance mapping for closely packed buildings in ALS point clouds. They divided point clouds into segments based on random sample consensus (RANSAC), by which points connecting different instances are believed to be eliminated. Separated segments are merged into building instances based on predefined rules. For example, two neighboring segments are merged into one object if their convex hulls intersect. The main problem of these methods lies in the definitions of rules, which may vary by scene. Also, it is difficult to find optimal stop criteria to terminate the segment-merging process if building instances are tightly connected [19].

The second group of methods is based on the shape priors of building outlines. Awrangjeb *et al.* [20] presented an individual building detection algorithm using both ALS data and aerial images. Their key idea to detect a building instance is to connect line segments derived from the input data (ALS and multispectral information) into a rectangle, i.e., one rectangle corresponding to one building instance. This assumption is simple but effective and has been adopted

Manuscript received December 8, 2020; revised March 31, 2021 and April 30, 2021; accepted May 28, 2021. Date of publication June 17, 2021; date of current version January 12, 2022. This work was supported in part by the Natural Science Foundation of China under Grant 42071443, in part by the Natural Science Foundation of Jiangsu Province under Grant BK20200784, and in part by the Natural Sciences and Engineering Research Council of Canada. *(Corresponding author: Ruisheng Wang.)*

Shaobo Xia was with the Department of Geomatics Engineering, University of Calgary, Calgary, AB T2N 1N4, Canada. He is now with the College of Environmental and Resource Sciences, Zhejiang A&F University, Hangzhou 311300, China (e-mail: shaobo.xia@qq.com).

Sheng Xu is with the College of Information Science and Technology, Nanjing Forestry University, Nanjing 210037, China (e-mail: xusheng@njfu.edu.cn).

Ruisheng Wang is with the Department of Geomatics Engineering, University of Calgary, Calgary, AB T2N 1N4, Canada (e-mail: ruiswang@ucalgary.ca).

Jonathan Li is with the Department of Geography and Environmental Management, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: junli@uwaterloo.ca).

Guanghui Wang is with the Department of Computer Science, Ryerson University, Toronto, ON M5B 2K3, Canada (e-mail: wangcs@ryerson.ca).

Digital Object Identifier 10.1109/TGRS.2021.3087159

in other building mapping studies. A marked point process model was applied to detect rectangular building instances from ALS point clouds [21]. Yang *et al.* [22] used a marked point process model to extract building outlines from ALS point clouds, replacing the rectangular model with 3-D cubes. The rectangle assumption was applied to group point clouds in individual building instances in mobile laser scanning data processing [23]. These methods have two limitations. The assumption is too strong to cover nonrectangular buildings in the real world and those whose outlines can be approximately represented by rectangles. Protruding parts (e.g., a porch) may be omitted by rectangular representations. These methods cannot separate spatially connected buildings [22], which are common in urban scenes.

In recent years, many deep learning frameworks have been proposed for point cloud registration, classification, semantic segmentation, and instance segmentation [24], [25]. The objective of instance segmentation is to identify individual objects at the point level. Thus, dividing ALS point clouds into building instances can be viewed as a specific task of instance segmentation. The existing instance segmentation methods can be categorized into proposal-based methods and proposal-free methods [26]. 3-D-BoNet [27] is a representative proposal-based instance segmentation method that first predicts 3-D bounding boxes for object instances, and then, pointwise labels are obtained via a classification branch. Occuseg [28] is a proposal-free network for the instance segmentation of point clouds. The key idea of this method is to utilize the occupancy information of each instance. It uses the UNet as the backbone network and combines feature, spatial, and occupancy terms to learn embedding representations. Finally, the instance-level segmentation results are obtained by an instance clustering algorithm. PointGroup [29] also uses the UNet architecture as the backbone network. The learned features are fed to a semantic segmentation branch and an offset prediction branch. After that, original and shifted point clouds are grouped into instances via a clustering module. The segmentation accuracy achieved by this method on an international benchmark ScanNetv2 is around 64%. Although there exist many frameworks for the instance segmentation of point clouds, these methods are mainly proposed for indoor point clouds and cannot be applied to outdoor large-scale point clouds directly, as discussed in [30]. In fact, deep learning frameworks are mainly applied to semantic segmentation in large-scale LiDAR point cloud processing [31]–[34]. Thus, the instance-level segmentation frameworks for large-scale outdoor point clouds still lag behind.

To sum up, the major challenge in previous studies lies in the ambiguous definition of building instances in point clouds. Compared with objectness measures based on various features (e.g., salience, edges, and color contrast) in 2-D images [35], there are far fewer clues in ALS point clouds. Also, buildings in the real world often have complex structures and are spatially connected. Therefore, it is difficult to give universal definitions of building instances in ALS point clouds. We do not aim to explore the definitions of building instances or propose sophisticated rules to group building points into individual objects. We propose a building instance mapping method for ALS point clouds aided by publicly accessible polygonal footprints from various data sources, such as OpenStreetMap (OSM).

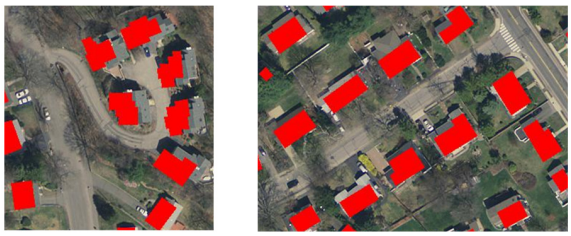


Fig. 1. Two typical errors of building footprints in OSM: (a) alignment errors between polygons (red) and real buildings and (b) shape differences between polygonal footprints and corresponding building masks.

The intuitive idea for footprint-aided instance segmentation is to clip points by projecting existing polygons onto ALS point clouds [36]. However, there are two significant problems with directly using existing footprints. First, significant alignment errors always exist among open-access building polygons [37], [38]. For example, there is a significant shift between online maps and real buildings, as shown in Fig. 1(a). In this situation, the offset polygons will lead to incorrect clips in point clouds, especially for connected buildings. The second is that building footprints may not have the same shapes as in remote sensing data, and outlines are often simplified [36]. For example, the outlines and sizes of several building polygons in Fig. 1(b) differ greatly from the real buildings. In this case, the extracted building instance following the polygonal guidance will be incomplete, and further refinements will be necessary. Thus, accurate matches are necessary between polygonal data and ALS point clouds before conducting instancewise point cloud clipping. However, in OSM data matching, most research has focused on matching OSM data to vectorial objects, such as cadastral maps [39]–[41], and there has been little study on matching OSM data to discrete and unorganized ALS point clouds. A method to register terrestrial point clouds to 2-D maps was proposed, so as to estimate the unique transformation matrix between point clouds and existing polygons by matching wall planes [42]. This strategy is impractical for ALS point clouds. The transformation matrices for building instances and their corresponding polygons often vary; this problem is likely to involve nonrigid rather rigid registration. Walls are always missing in ALS point clouds, making it impossible to initialize corresponding elements required in this method. Therefore, new matching methods are needed for ALS point clouds and polygonal footprints.

The meaning of instance-level building segmentation via matching footprint polygons to ALS point clouds can be illustrated from three aspects. First, as the definition of building instances in the real world remains unclear, utilizing existing footprints is an alternative and effective way to extract individual objects from ALS point clouds. Second, existing building footprints can be updated and improved if existing polygons and building instances in ALS point clouds are matched [43]–[45]. Third, aligning footprint maps with ALS

point clouds at the instance level is important for preparing accurate training datasets in deep learning research. For example, training datasets for the semantic segmentation of remote sensing images were made based on OSM footprints [46], [47]. Thus, the alignment of OSM data and remote sensing images has attracted more attention [48]–[50]. However, this related research is proposed only for remote sensing images under deep learning frameworks.

Overall, little research has been done in the field of footprint-aided point clouds processing, and all the related methods cannot handle spatially adjacent or connected building instances [51]. In this work, we are not trying to make incremental progress based on the existing building instance mapping frameworks. In fact, we aim to achieve instance-level building segmentation results in ALS point clouds, guided by the widely accessible polygonal footprints. The proposed new solution does not rely on specific definitions of building instances and can deal with the connected and adjacent buildings. Our main contribution is twofold.

1) We propose a new framework to extract building instances from ALS point clouds. In our solution, individual building candidates are generated by merging point clouds automatically in a bottom-up manner. The merging process needs no specific rules or oversimplified assumptions, which most of the existing 3-D building mapping methods heavily rely on.
2) We present a robust matching model to register 3-D building candidates and existing 2-D polygonal footprints, while most of the existing matching methods can only handle 2-D images, 2-D vectorized data, or 3-D point clouds individually. The proposed model enables the instance extraction in challenging scenes, such as spatially connected and adjacent buildings, which all the existing methods cannot.

Also, we find an L0 gradient minimization-based smoothing method [52] helpful in separating connected flat roof planes. Besides, our work also explores the potential of existing polygonal maps and can help in producing datasets for deep learning researches.

The rest of this article is organized as follows. Section II presents our proposed methods to merge segments and achieve instance-level roof extraction. Section III discusses our experiments, parameter tuning, and comparison of methods. Section IV summarizes our contributions, challenges, and future work.

II. Method

The input of our framework is building regions in ALS point clouds, which can be identified by many methods [3], [16], [53]. We remove ground points from the original ALS data via the cloth simulation-based algorithm [54]. Building regions in the remaining data are further extracted by a graph-cut aided method [53].

Fig. 2 shows a flowchart of our method. The basic idea is to merge roof segments into instances. The input building points are segmented by applying region growing [36]. Connected and flat roofs that cannot be separated are divided into pieces by smoothing and resegmentation. A graph of segments is constructed based on their spatial adjacency relationships, and candidates of building instances are generated according to the segment graph. A new method to match building candidates and polygonal footprints is proposed to divide ALS point clouds into building instances.

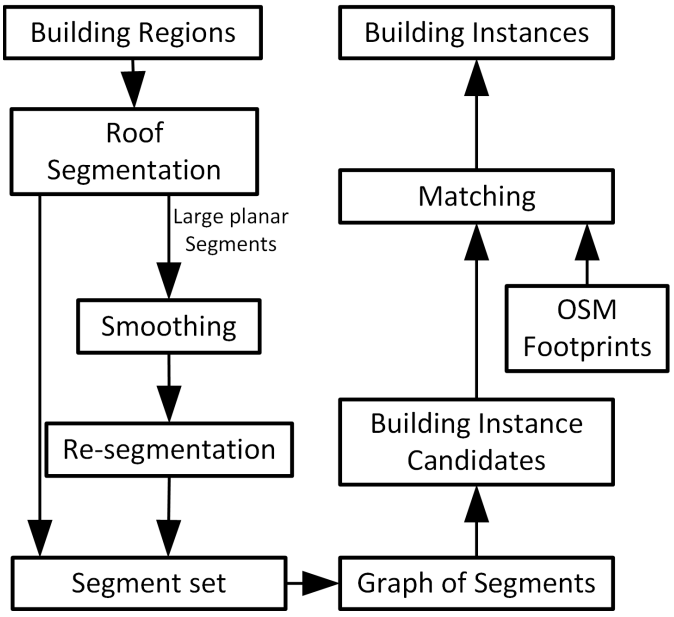


Fig. 2. Flowchart of the proposed building instance mapping method.

A. Segmentation of Planar Roofs

Roof segmentation in ALS point clouds has been widely studied [36], [55]–[57]. In this study, input building points are divided into segments by a region growing-based method [36]. However, this method will fail to separate spatially connected flat roofs, as shown in Fig. 3. The initial segments whose width is large (e.g., $\geq$10.0 m) will be further segmented in the following steps. As these roofs are always at different heights, it is reasonable to use the region growing method with the growing criterion on the height differences (dHs) between neighboring points. However, the dHs of adjacent roofs are often small, and the growing process is easily affected by noisy points. Thus, it is necessary to smooth these roof segments before further segmentation.

We find the L0 gradient minimization method [52], [58] suitable, as it is good at preserving edges in the original data. The core idea of this smoothing method is expressed as

$$\min_{R} |R - \hat{R}|^2 + \lambda \cdot G(R) \tag{1}$$

where $\hat{R}$ is the input noisy data, R is the expected smoothed data, $G(R)$ is the gradient after smoothing, and λ controls the importance of the gradient term. The gradient differences between neighbors, as well as the differences between $\hat{R}$ and R, will generally be minimized if an optimal R is found. In our application, planar segments are converted to 2-D and smoothed by this method and then a region growing algorithm that thresholding on dHs is further applied. Fig. 3 shows five

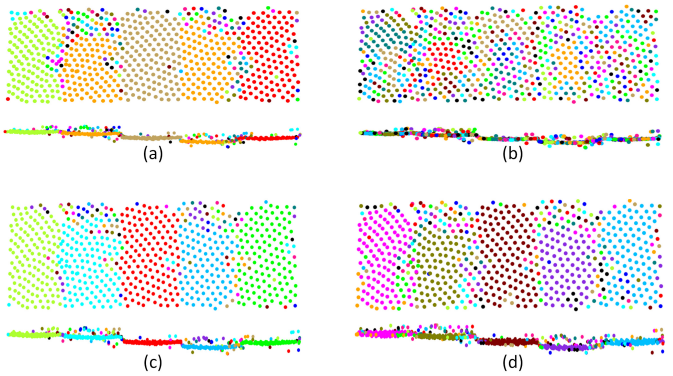


Fig. 3. Flat roof resegmentation results: (a) after applying smoothing ($dH = 1$ cm); (b) without smoothing ($dH = 1$ cm); (c) after smoothing ($dH = 2$ cm); and (d) without smoothing ($dH = 2$ cm). In each subfigure, the upper part is the top view of the data, and the bottom part is the side view.

independent roofs to be separated. Fig. 3(a) and (b) shows the segmentation results for smoothed and unsmoothed data, respectively, using $dH = 1$ cm. The results in Fig. 3(a) are clearly much better than those in Fig. 3(b), where many small segments are detected. By increasing dH to 2 cm, the segmentation results of smoothed [see Fig. 3(c)] and unsmoothed [see Fig. 3(d)] data are close, while detected segments in Fig. 3(c) are still larger than those in Fig. 3(d). One significant advantage of imposing smoothing before segmentation is that we can apply a small threshold to obtain satisfactory results while avoiding oversegmentation.

B. Generation of Building Candidates

Individual objects in images can be formed by hierarchically merging segments [59]. Hence, we propose a segment merging algorithm to obtain building candidates from previously extracted segments. This can be explained through a toy example. Suppose that there are two building instances, as in Fig. 4(a), and four segments are detected, as shown in Fig. 4(b). The expected merging result should contain two instances. The first is A ∪ B ∪ C, which corresponds to building instance 1, and the second is D.

We define the number of segments to be merged for one instance as the merging degree K. As merged instances may have various K values, there is no feasible prior knowledge to fix K in advance. To solve this problem, we enumerate all possible combinations of these segments with different k values and try to select a subset of these combinations as potential instances. The set of building instance candidates is defined as

$$S = \{S_1, S_2, \ldots, S_k\} \tag{2}$$

where S_k represents segment combinations when K equals k. If there are n roof segments, then there will be $\binom{n}{k}$ combinations for a given k. It is obvious that many combinations, such as A ∪ B ∪ D in this example, cannot be grouped into one object, as their segments are not spatially connected. To remove these combinations, we construct a graph of segments, as shown in Fig. 4(c). If the minimum distance between two sets of points is small (e.g., 5.0 m),

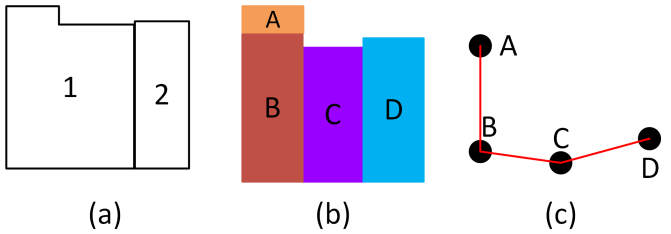


Fig. 4. Toy example of segment merging: (a) outlines of two building instances; (b) four segmented roofs in point clouds; and (c) graph of adjacent map for (b).

$$Adj = \begin{array}{c} \begin{matrix} A & B & C & D \end{matrix} \\ \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{matrix} A \\ B \\ C \\ D \end{matrix} \end{array} \qquad Adj^2 = \begin{array}{c} \begin{matrix} A & B & C & D \end{matrix} \\ \begin{bmatrix} 2 & 2 & 1 & 0 \\ 2 & 3 & 2 & 1 \\ 1 & 2 & 3 & 2 \\ 0 & 1 & 2 & 2 \end{bmatrix} \begin{matrix} A \\ B \\ C \\ D \end{matrix} \end{array}$$

(a) (b)

Fig. 5. Demo for the adjacency matrix based on the example in Fig. 4. (a) Adjacency matrix. (b) Second power of the adjacency matrix.

then an edge will exist between them in the graph. After that, the adjacency matrix Adj will be constructed based on the graph. It should be noticed that the matrix diagonal elements of Adj are 1, which is different from the classical adjacency matrix, where these elements should be 0. To check whether segments in the graph are connected, we adopt the power of the adjacency matrix [60]. If all elements in the matrix are greater than zero, then this group of segments is labeled as connected instances. Otherwise, they are not spatially connected.

For example, an adjacency matrix derived from the graph in Fig. 4(c) is shown in Fig. 5(a). If two segments are adjacent, the corresponding elements in the adjacency matrix are set to 1; otherwise, the elements will be set to zero. As only spatially connected segments can form building instances, invalid building candidates can be removed in advance before the model optimization. This validation process can be done based on the adjacency matrix. Specifically, segments A and B are first-order neighborhoods; thus, the elements indexed by (1, 0) and (0, 1) in the matrix [see Fig. 5(a)] are larger than 0. On the other hand, segments A and C or B and D cannot form valid candidates as their corresponding elements in the matrix are zero. After raising the adjacency matrix to the second power, the second-order neighborhood is considered in the model validation. As shown in Fig. 5(b), the elements indexed by (2, 0) and (0, 2) in this matrix are larger than zero, indicating that segments A and C can be grouped into one candidate now. In this case, the elements indexed by (0, 3) and (3, 0) still equal zero, meaning that segments A and D cannot appear simultaneously in one building candidate.

C. Matching Model

Given a set of instance candidates and polygonal footprints in a region, the instance-level building segmentation task can be solved by finding the best matches between candidates and footprints. Fig. 6 shows an example of this method. The original roof points in Fig. 6(a) are segmented, as shown

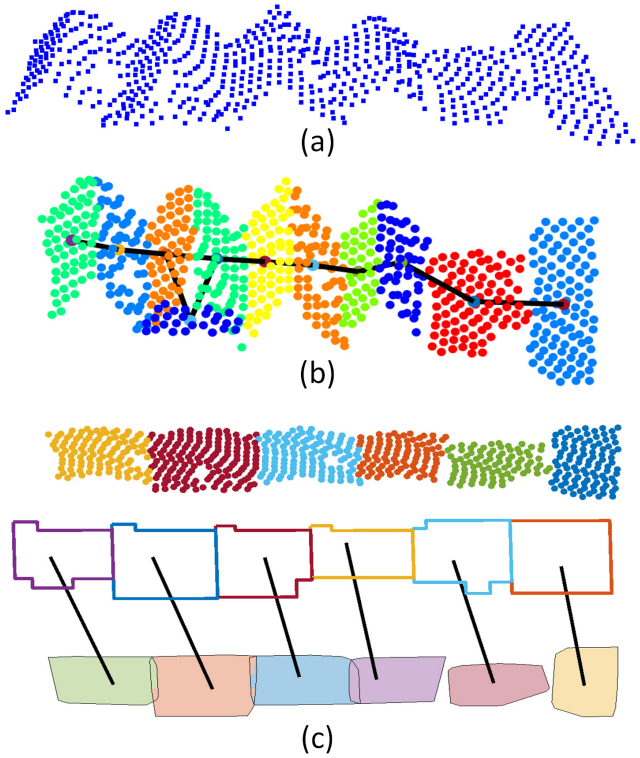


Fig. 6. Example of instance-level segmentation by our matching method. (a) Input roof points. (b) Segmented roofs are colored differently, and the corresponding adjacent graph (black lines). (c) Our matching results, where extracted instances in the first row are colored differently. Polygonal footprints are colored randomly, as shown in the middle. The bottom shows polygonal instances corresponding to extracted buildings in the first row. Black lines linking polygonal footprints and instances indicate matching relationships.

in Fig. 6(b). A set of building instance candidates can then be generated by applying the method described in Section II-B. Suppose that there are m candidates in S and a set (Y) of building polygons. To form initial matches, all the building candidates in S and the input polygonal footprints in Y are coarsely paired. Specifically, the z-nearest polygon neighbors of each candidate in S are used to form potential matches. A small z (e.g., 3) works for all the experiments in our work. The lth match p_l represents $\{x_i \leftrightarrow y_j\}$, where x_i is the ith candidate in S, and $\leftrightarrow$ means that x_i and one polygon y_j in Y are paired. Then, the instance-mapping task equals finding the optimal subset from all the initial matches.

An objective function, as shown in the first line in (3), is proposed to find optimal matches from potential matching pairs. In this model, p_l and x_i are binary variables. If a potential match is selected, then p_l equals 1, and otherwise, it is 0. $N^l(x_i)$ is the number of segments in an instance candidate x_i, and x_i forms a potential match p_l. By maximizing the objective function, we cover as many roof segments as possible. $A^l(x_i, y_t)$ is calculated based on (4), where $S(x_i, y_j)$ is the overlapping area between the instance candidate (x_i) and the polygonal footprint (y_t). $S(x_i)$ and $S(y_i)$ are areas of polygonal regions of x_i and y_t, respectively. If the overlapping area is large, then $A^l(x_i, y_t)$, which measures the similarity between polygonal footprints and instance candidates, will be large. It should be noticed that big spatial shifts may exist between instance candidates and footprints, as shown in Fig. 7(a). Thus, a paired polygon should be translated to make centroids of x_i and y_t overlap before calculating $A^l(x_i, y_t)$

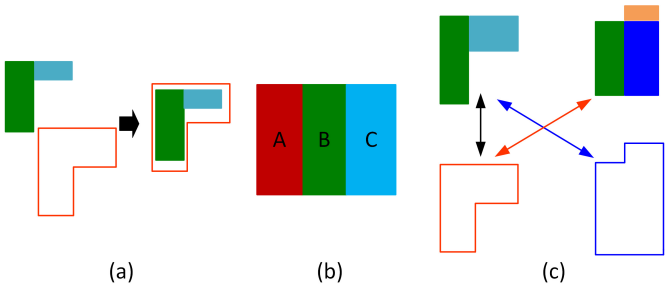


Fig. 7. Constraints in (3). (a) Translating polygons before calculating overlapping areas. (b) Two instance candidates formed by {A, B} and {B, C} cannot be selected simultaneously. (c) Examples of conflicted matches.

$$\begin{aligned} &\max \sum_{l=1}^{n} (N^l(x_i) \cdot p_l + A^l(x_i, y_t) \cdot p_l) \\ &\text{s.t. } \forall\, p_l = \{x_i \leftrightarrow y_t\} : p_l \le x_i \\ &\quad \forall\, \{i, j\},\ \text{if } x_i \cap x_j \ne \emptyset : x_i + x_j \le 1 \\ &\quad \forall\, \{l, m\},\ \text{if } p_l \otimes p_m : p_l + p_m \le 1 \\ &\quad p_l \text{ and } x_i \text{ are binary variables.} \end{aligned} \tag{3}$$

$$A^l(x_i, y_t) = \frac{S(x_i, y_t)}{S(x_i)} \cdot \frac{S(x_i, y_t)}{S(y_i)}. \tag{4}$$

Directly maximizing the objective function in (3) will result in the selection of all n paired matches. Thus, the objective function should be constrained before optimization. There are three constraints. The first is $p_l \le x_i$, i.e., if a building candidate x_i is not selected ($x_i = 0$), then the corresponding matching pair p_l should also be discarded ($p_l = 0$). The second is that two building candidates that share the same segments, as shown in Fig. 7(b), cannot be selected simultaneously, that is, given two overlapping instance candidates x_i and x_j, $x_i + x_j \le 1$. The last constraint concerns conflicts between n paired matches. Two situations are of concern. The first is that two potential matches should not contain the same objects. In Fig. 7(c), a double-headed arrow represents a potential match. In this case, the matches in black and red should not be selected at the same time. Neither should the black and blue matches are selected simultaneously, as both contain the red polygon. The second situation occurs when matches cross, such as the red and blue ones in Fig. 7(c). We introduce the $\otimes$ symbol to indicate these two situations. This last group of constraints can be expressed as "if $p_l \otimes p_m : p_l + p_m \le 1$" for two conflicted matches p_l and p_m. The constrained matching pair selection model is expressed as (3). It is a linear programming problem that can be solved by a branch and bound algorithm [61].

III. Experiments and Analysis

The proposed method was implemented in MATLAB and tested on a personal computer with 16-GB RAM and an i7-6700 CPU. We first discuss large-scale experiments conducted on two typical scenes. We then focus on challenging regions where building instances are spatially connected.

We analyze the robustness of our method, and then, we present the selection of parameters, comparisons, discussion, and limitations.

A. Large-Scale Experiments

Two typical sites from the publicly available DataPlus dataset [47] were used in our experiments. The first site (see Fig. 8) is in Arlington, MA, USA, which is a typical suburban area. This site covers 540 m × 470 m and contains 1 215 060 points. The second site (see Fig. 9) is a typical urban scene in San Francisco, CA, USA. This site has 129 651 ALS points and covers a rectangular region that is 410 m × 280 m. The point density in the Arlington dataset is 4.8 pt/m^2, and the point density in the San Francisco dataset is 1.2 pt/m^2. The ALS point clouds were collected by state and federal agencies, with an average point spacing of around 0.5 m. The polygonal building footprints were downloaded from the OSM. As shown at the beginning of Section II, the ground points were removed by a cloth simulation-based filtering method [54], and building regions were extracted by a graph-cut-based method [53] in preprocessing. To focus on our proposed method, significant errors (e.g., points of one building instance are all mislabeled as vegetation) are manually revised after preprocessing.

As polygonal footprints in these two sites are not largely shifted from ALS point clouds, based on the conclusion that most of the mean shifted error in OSM is within 4.0 m [37], we added noise to OSM polygons to simulate difficult and complicated situations. Specifically, we added uniformly distributed errors in the range $[-e, e]$ to X and Y components of polygonal footprints, where the error e was set to 6.0 m at these two sites, as the positioning errors of 90% of footprints are below 6 m [37]. We found a value of 4 to be suitable for the merging degree K at these two sites, and building segments not merged into any instances were assigned to the closest buildings. As shown in Figs. 8 and 9, our proposed method can achieve instance-level segmentation results in large-scale scenes.

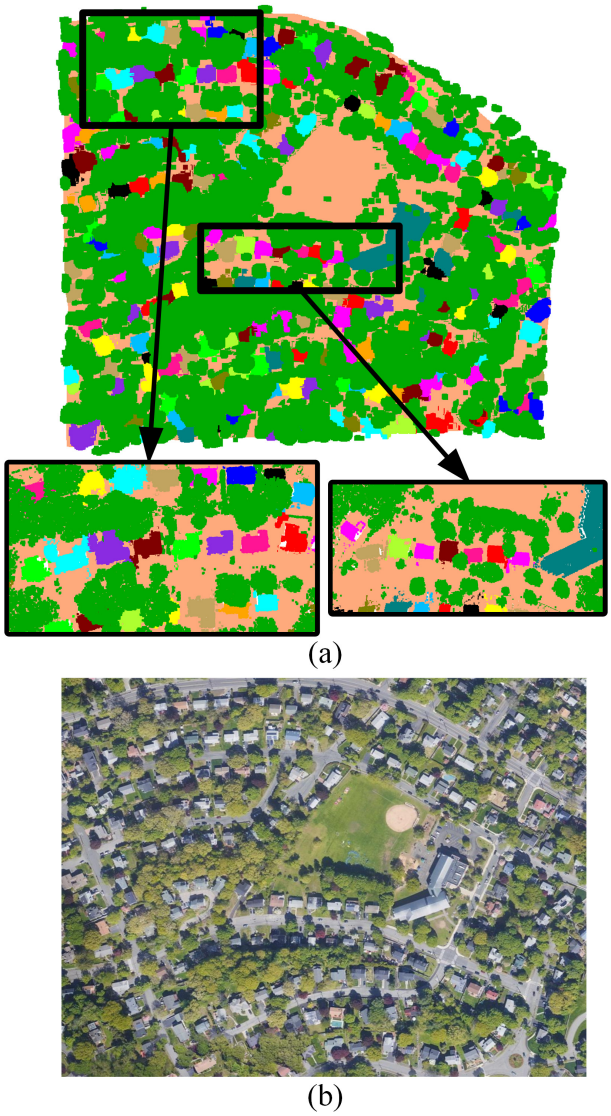


Fig. 8. Instance-level building mapping results of the Arlington site. (a) Mapping results are colorized. Ground points are ochre; vegetation is green. Segmented building instances are colored randomly. (b) Corresponding image of this site from Google Maps.

B. Tests and Evaluations

As buildings in suburban areas (Arlington) are spatially separated, polygonal footprints can often be assigned to their nearest roof clusters, in which case our method is not required. Thus, we focus on connected buildings. Three scenes were extracted from the San Francisco site, as shown in Fig. 10. There are 12 instances in scene I, 17 in scene II, and 23 in scene III. These are all closely connected and cannot be vertically separated. Following the procedure in Fig. 2, roof points were segmented into small regions and merged to make potential instances. By finding optimal matches between instance candidates and polygonal footprints, roof points were grouped into independent instances, as shown in Fig. 10. The optimization of the matching model consumed little time in these scenes. For instance, it took less than 0.02 s to solve the model for scene III.

To demonstrate the robustness of our p method with positioning errors in polygonal footprints, we added pointing errors (e) to polygons, with the range of 0–10 m. In Fig. 10, the segmentation performances when e equals 0.0, 6.0, and 10.0 m are shown from left to right. Precision, recall, and F1 measure are introduced to our quantitative analysis and are defined as follows:

$$\text{Precision} = \frac{\text{TP}}{N_{\text{Seg}}} \tag{5}$$

$$\text{Recall} = \frac{\text{TP}}{N_{\text{GT}}} \tag{6}$$

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{7}$$

where TP is the number of correctly segmented instances, N_{Seg} is the number of segmented instances, and N_{GT} is the number of ground-truth instances. The values of precision, recall, and F1 measure change with position error e and are drawn in Fig. 11. This graph demonstrates that, with small shifted errors (e.g., less than 3 m), our method can produce satisfying results, in which all instances are correctly segmented. As the

shifting error increases from 3.0 to 10.0 m, segmentation accuracy drops from 1.0 to about 0.6. When the shifted error is 6.0 m, the segmentation accuracy is close to 0.9, which indicates that the method can handle most positioning errors in polygonal footprints. Examining the details, wrong matches are mainly caused by selecting wrongly merging instances. If segments from more than one building instance are merged into one object, then some polygonal footprints may not be matched to any instances. This explains why recall is less than the precision in our experiments. There is a big drop in accuracy when the shifted error increases to 10.0 m., mainly because the initial potential matches may not contain all correct pairs if the position error is too large.

To further illustrate the robustness of our method against position errors, Fig. 12 shows a close view of the segmentation results of scene II. There are ten building instances in this demo, in which all polygons are correctly matched to building instances. Although some polygonal footprints overlap and are cluttered, our method can find their best matches and extract independent roofs in this challenging scene.

C. Selection of K

The merging degree K is a key parameter in our method, which controls the maximum number of segments that an instance candidate can have. As buildings in suburban sites are more complex in terms of structures than those in urban areas, a subset of buildings at the Arlington site was extracted to analyze the selection criterion for K. We changed K from 1 to 6 in our experiments and recorded the running time, the number of candidates, and matching accuracy, as shown in Fig. 13. Similar to the configuration in the large-scale test, random position errors ($e = 6.0$ m) were added to polygonal footprints.

The major differences when K changes from 1 to 6 are in the running time and the number of instance candidates, while the matching accuracy in this scene remains at 1.0. Most computing cost is in guess generation and checking connections between segments, as shown in Section II-B. As shown in Fig. 13(a), the time consumed does not change significantly until K is 6. The number of potential instance candidates in Fig. 13(b) shows an upward trend as K increases from 1 to 4 and remains stable as K changes from 5 to 6. In Fig. 13(c), segments to be merged are colored randomly. The segmentation results in Fig. 13(d)–(f) correspond to $K = 1$, 4, and 6, respectively. It is clear that all building instances are correctly extracted, while some are not complete when K is small. For example, many small segments are missed when $K = 1$, as shown in Fig. 13(d). It should be noticed that the merging results in Fig. 13(e) and (f) are very close to each other although a large K is applied in Fig. 13(f). Based on these experiments, $K = 5$ reaches a good balance between time cost and segmentation accuracy in this complex scene, and for building instances with simple structures (e.g., in the San Francisco site), $K = 3$ is enough in most cases. Notice that, in incomplete instances, we can assign missing segments to their nearest building instances after matching.

It should be noticed that the proposed method has no limitation on the K value, which means that K can be larger than 6 at the cost of more computational resources. It is interesting to discuss how to handle very complicated roofs with a small K. Based on our analysis, there might exist two solutions. The first one is ignoring small roof segments when generating building candidates from original planar segments. By doing so, a small K value will still work for all buildings. The second solution may lie in merging small roof segments into larger ones before constructing the graph of segments. Thus, the max merging degree can also be reduced for extremely complex roofs.

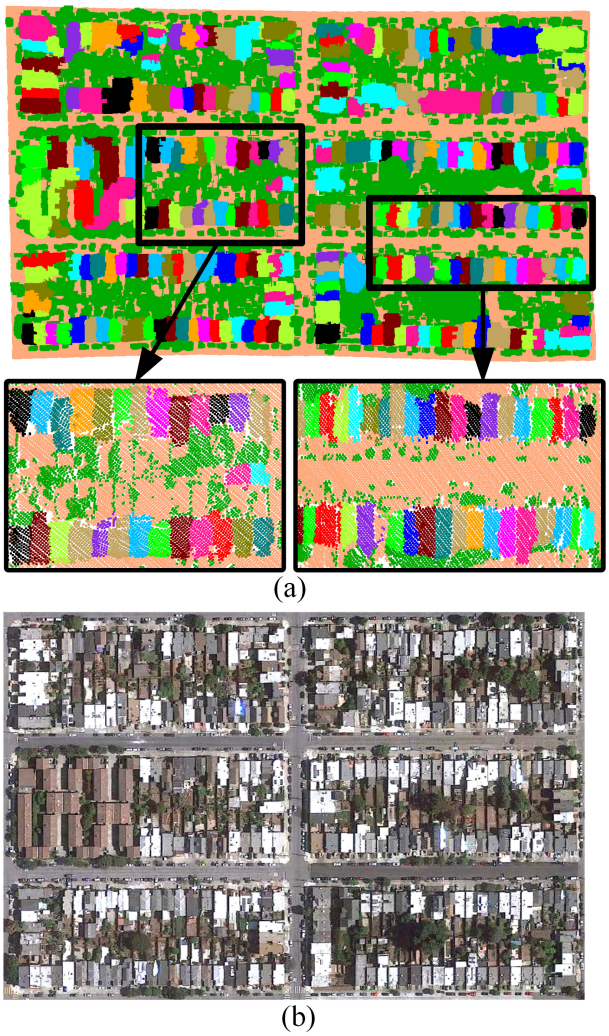


Fig. 9. Instance-level building mapping results of the San Francisco site. (a) Mapping results are colorized. Ground points are ochre. Vegetation is green. Segmented building instances are colored randomly. (b) Corresponding image of this site from Google Maps.

D. Discussion

1) Method Comparison: The aim of this work is to segment ALS point clouds into independent buildings. As reviewed in Section I, most of the existing studies focus on building regions instead of building instances. There exist few studies related to ALS building instance mapping, but none of them can handle complex scenes, such as connected buildings. Thus, the widely accessible polygonal maps (e.g., OSM) are utilized for extracting building instances from ALS point clouds.

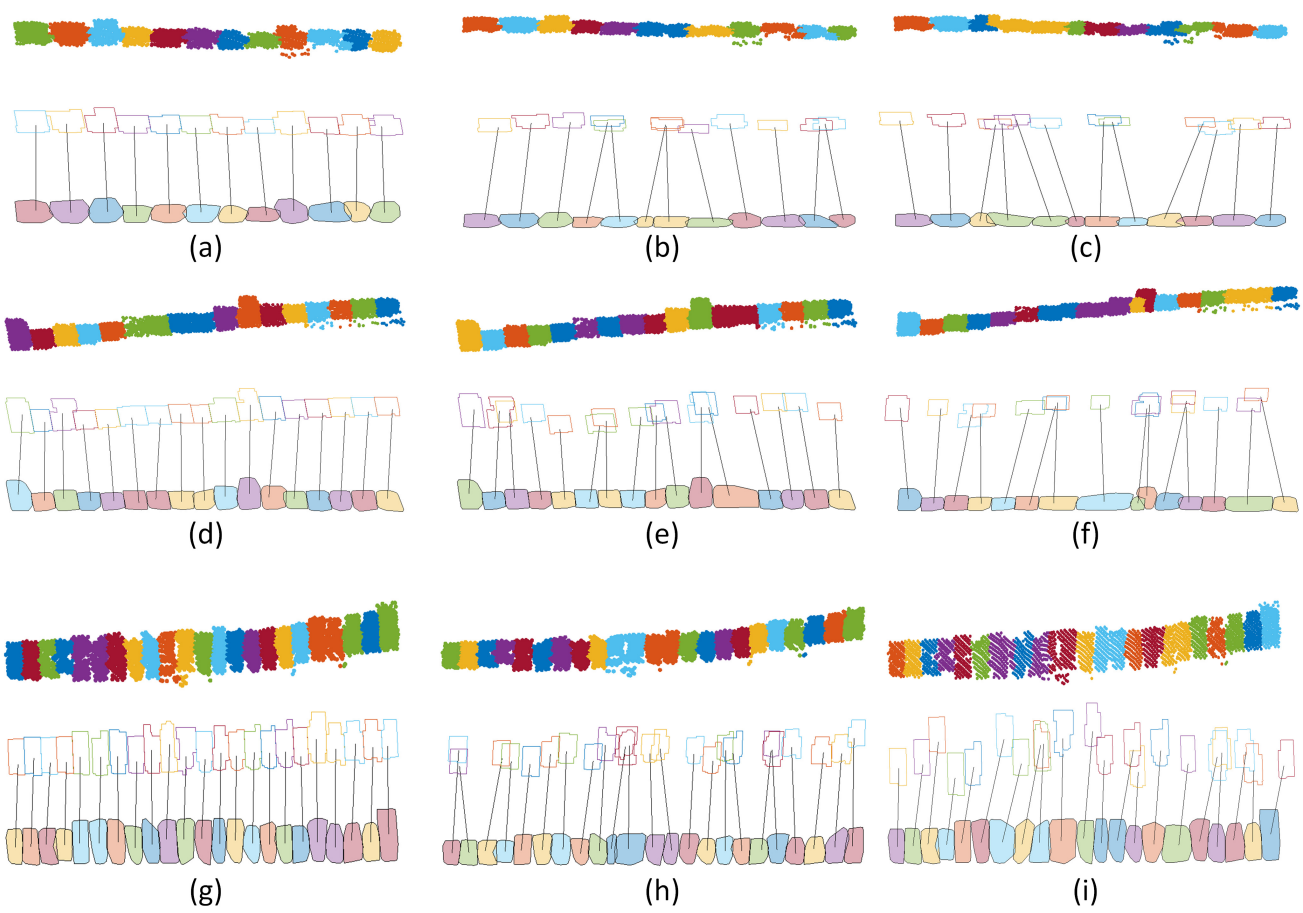


Fig. 10. Instance-level segmentation results in three scenes. Three rows correspond to three scenes. Three columns from left to right correspond to different position shifting errors. (a) Scene I, $e = 0.0$ m; (b) Scene I, $e = 6.0$ m; (b) Scene I, $e = 10.0$ m; (d) Scene II, $e = 0.0$ m; (e) Scene II, $e = 6.0$ m; (f) Scene II, $e = 10.0$ m; (g) Scene III, $e = 0.0$ m; (h) Scene III, $e = 6.0$ m; and (i) Scene III, $e = 10.0$ m.

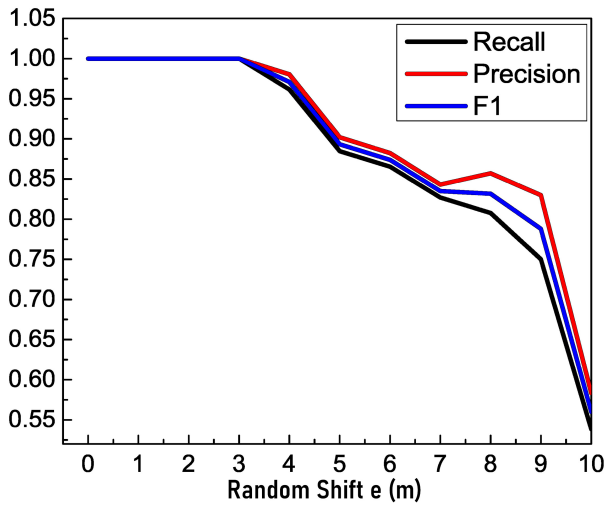


Fig. 11. Quantitative evaluations of instance-level building segmentation for different position errors.

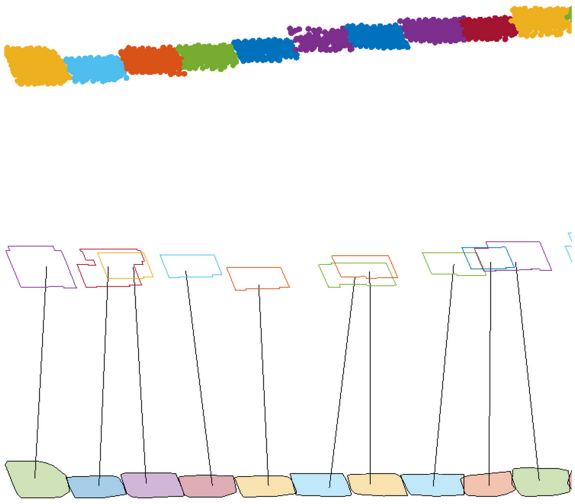


Fig. 12. Close view of instance-level segmentation results when e is set to 6.0 m.

It is interesting to compare our method with the existing polygon-based object mapping methods.

The polygon-based object mapping researches can be roughly summarized into three groups. The first one is matching OSM polygons to vectorized objects [40]. However, it is not an easy task to extract accurate and topological-correct building outlines from ALS point clouds. Also, spatially adjacent or connected building instances cannot be separated while reconstructing building outlines by modeling frameworks [62]. Thus, this type of method can only handle vectorized data and cannot be applied to discrete point clouds directly.

The second group is aligning the extracted building outlines from remote sensing images to the existing building polygons.

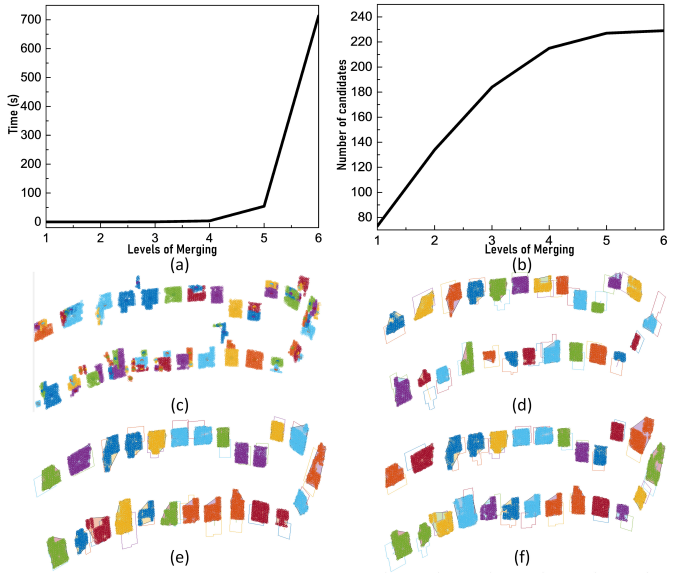


Fig. 13. Selection of K: (a) time cost when increasing K from 1 to 6; (b) number of instance candidates when increasing K from 1 to 6; (c) segments colored differently in a residential region; (d) instance extraction results when $K = 1$; (e) instance extraction results when $K = 4$; and (f) instance extraction results when $K = 6$.

The deep learning model is applied to predict building masks and displacements between segmented buildings and existing polygons [48]. However, this deep learning framework is proposed for remote sensing images that contain plentiful spectral and textural information that ALS point clouds lack. Also, these proposed deep learning frameworks cannot handle irregular and unordered ALS point clouds.

The objective of [42] is the closest to ours, which matches the existing polygons to point clouds captured by terrestrial laser scanning. However, their method cannot be applied to ALS point clouds. Their proposed method heavily relies on wall planes, which can be viewed as edges of polygonal footprints after the data projection. Yet, most ALS data lack wall points. Besides, the proposed method in [42] follows the rigid-body registration strategy, which uses a few paired planes to estimate the registration parameters for the whole scene. However, as shown in Fig. 12, building polygons may have various displacement values, making the polygon matching problem in ALS point clouds more likely to be a nonrigid matching one that the framework in [42] cannot handle.

Overall, the existing polygon-guided building instance mapping methods are proposed for vectorized data and remote sensing images, not discrete point clouds. Furthermore, the matching between polygons and roofs in ALS point clouds is a nonrigid registration problem instead of the rigid-body registration one. Compared with the existing methods, our proposed framework deals with discrete ALS point clouds, and different building instances are allowed to have different matching parameters.

To further demonstrate the necessity and advantages of our proposed method, we compare our method to the direct clipping method (DCM) that segments points into instances by direct clipping guided by polygonal footprints. Two typical demonstrations are introduced for comparisons. The first

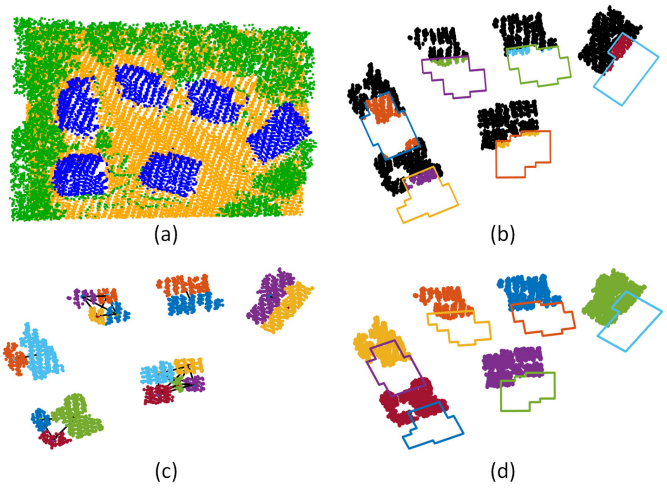


Fig. 14. Comparisons of DCM and proposed method in residential area. (a) Input building points are colored blue. (b) DCM results. Unassigned points are black. Points belonging to different instances are colored differently. (c) Input segments are organized by six adjacent graphs. (d) Segmented results achieved by our method. Different colors indicate various instances.

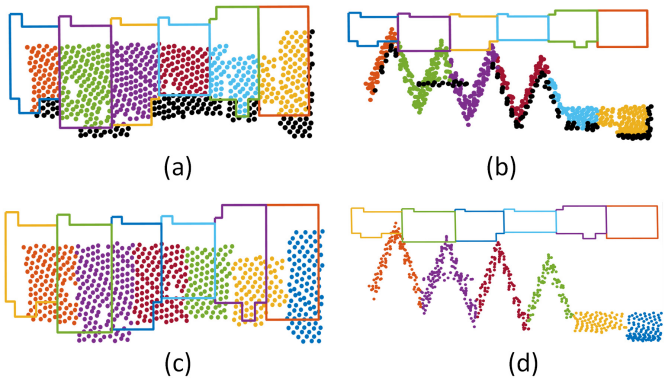


Fig. 15. Comparisons of DCM and the proposed method in the urban area. (a) Top-down view of DCM results. Unassigned points are black. Different colors indicate various instances. (b) Side view of DCM results. (c) Top-down view of results of the proposed method. (d) Side view of results of the proposed method.

contains six building instances that are not spatially connected, as shown in Fig. 14. The second also contains six instances, which are connected, as shown in Fig. 15. Polygonal footprints are all biased in these two demos.

In the first demo, the results extracted by DCM are shown in Fig. 14(b). It is clear that segmented building instances by DCM are biased and even contain points from adjacent buildings. If labeled (black) points in Fig. 14(b) are assigned to their nearest neighbors, then many extracted instances will contain points from other instances. Segments and OSM-guided merging results are shown in Fig. 14(c) and (d), respectively.

These figures clearly demonstrate that segment-based merging and matching are not sensitive to positioning errors. In more challenging cases, such as the example in Fig. 15(a) and (b), the position errors of polygonal footprints still lead to large errors when applying DCM. The proposed method still achieves satisfying results, as shown in Fig. 15(d) and (e). Comparison of these two demos clearly demonstrates the superiority of our framework.

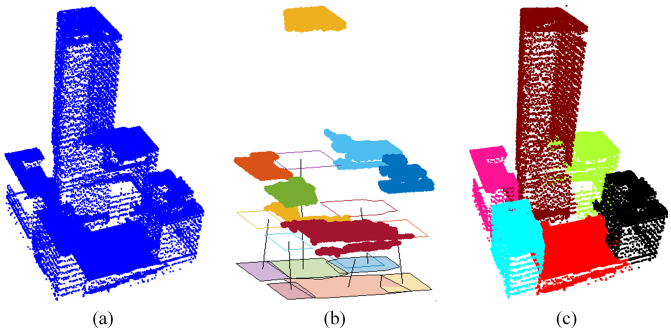


Fig. 16. Building instance mapping in high-rise building blocks. (a) Original point clouds. (b) Matched roof candidates. (c) Final instance mapping results.

2) Robustness Analysis: Based on the experiments and analysis, the difficulty of mapping building instances with polygons is mainly determined by the similarity between adjacent buildings. The more similar the structures of adjacent buildings, the more difficult the instance mapping is. In the previous experiments, low buildings are chosen for various tests due to the high roof similarity among neighbors in these datasets. Specifically, the geometric shapes, sizes, and elevation of roof components of these buildings are quite close. The geometric shapes of roof components may still be similar for high-rise buildings, but their sizes and elevations are always different. Thus, compared with low and connected buildings in the previous experiments, separating connected high-rise buildings is often less challenging. To further verify this idea, the proposed framework is also tested in point clouds of high-rise buildings.

As shown in Fig. 16(a), the original ALS point clouds of high-rise buildings are colored blue. This dataset is made up of around 100k points and consists of six connected building instances. The tallest one in the middle is about 120-m high, which is surrounded by five building instances whose heights range from 12 to 40 m. After applying the roof segmentation and merging, six roof candidates are generated and matched with the existing building polygons, as shown in Fig. 16(b). The matched roof instances are colored differently, and black lines indicate the matched pairs. Finally, the original point clouds are assigned to their closest roof instances. As shown in Fig. 16(c), the input building block is segmented into six building instances. This test demonstrates that the proposed building instance mapping framework is also applicable in high-rise buildings.

The previous tests mainly focus on the most common shifting errors in polygonal footprints. However, other polygon-related errors, such as data redundancy and data missing, may also exist. The data redundancy means that the polygon number is larger than the number of building instances to be matched. In this case, it is possible to obtain oversegmented buildings based on the proposed framework. For example, as shown in Fig. 17(a), there are six building instances in point clouds, while seven building polygons are used in the matching process. Based on the proposed framework, five building instances in point clouds are segmented correctly. However, the leftmost building instance is oversegmented into two objects corresponding to the two polygons on the left.

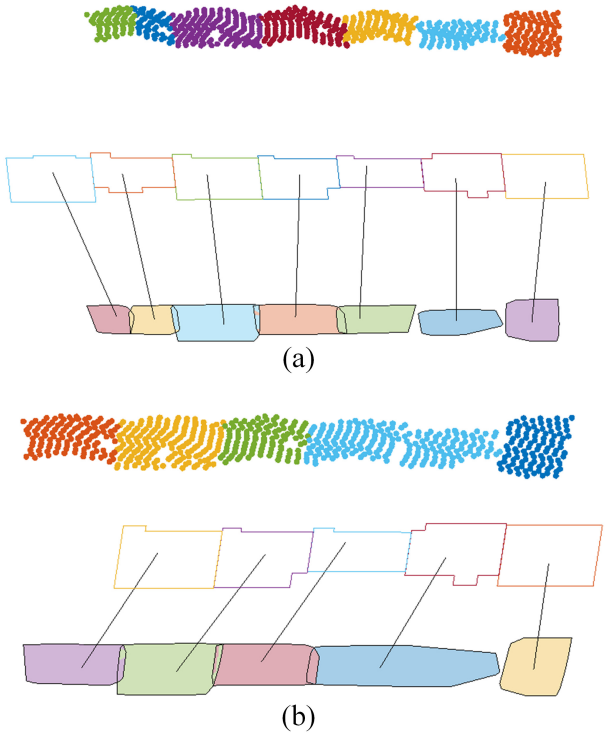


Fig. 17. Roof matching examples for footprint redundancy and missing. (a) Example of data redundancy. (b) Example of polygon missing.

On the other hand, if some of the building polygons are missing, the matching process may fail. An example of the polygon missing is shown in Fig. 17(b). There are five building polygons and six building instances in this demo. The results show that only one polygon on the rightmost is correctly matched to the building instance in point clouds, while the rest four polygons are paired mistakenly. Although the matching performance is poor, four out of six building instances are segmented correctly. The reason for this phenomenon is that the adjacent building polygons are quite similar, and thus, the mismatched pairs can still provide proper guidance for the roof merging. Overall, beyond the common shifting errors that are deeply studied in this work, other footprint errors, such as data redundancy and data missing, will also influence the instance segmentation accuracy. However, the mismatches between polygons and roofs may not always lead to wrong mapping results.

E. Challenges

Although the effectiveness and robustness of our proposed method have been demonstrated, there still exist two challenges. First, our method relies on the quality of data preprocessing, such as building point classification. For instance, vegetation between houses is misclassified as buildings and assigned to detected building instances in Fig. 18(a). This may not affect instance-level matching, but it will affect postprocessing, such as map updates. Second, polygonal footprints and building instances in ALS points often have quite different shapes because polygonal footprints are often simplified, and OSM footprints and ALS data are often collected at different

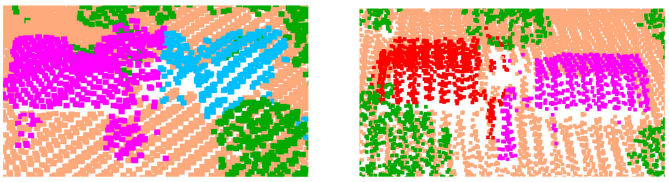


Fig. 18. Problems in instance-level building matching. (a) Misclassified vegetation points are assigned to building instances. (b) Enclosures and protruding parts of buildings can be wrongly assigned.

times, during which building outlines may change. For example, protruding parts between two houses [see Fig. 18(b)] may not be contained in polygons. This may affect matching, as the polygonal guidance is not consistent with ALS data.

Last but not least, we think that the ideal way for tackling this challenge is to extract building instances from point clouds directly. Currently, the state-of-the-art framework for the point-level instance mapping is based on deep learning, and the instance segmentation accuracy in the indoor benchmark is around 70% [29]. Furthermore, most of the existing instance mapping frameworks are proposed for indoor point clouds and cannot be directly applied to large-scale outdoor point clouds due to the massive data volume and complex scenes [30]. In short, there is still a long way to go before getting high instance mapping accuracy (over 90%) in large-scale point clouds. Therefore, the proposed framework provides a practical solution for this challenging task. Besides, large amounts of labeled point clouds at the instance level are the basis for deep learning studies. However, preparing the ground truth is both time-consuming and labor-intensive, and the existing outdoor benchmarks are only labeled for semantic segmentation [26]. Thus, the proposed framework may also help in preparing the instance-level benchmarks of ALS point clouds.

IV. Conclusion

A small number of studies have been conducted in the extraction of building instances from ALS point clouds. As the existing methods always need specific rules or empirical assumptions, they cannot achieve high instance-level mapping accuracy in large-scale datasets or deal with complicated scenes, such as spatially connected or adjacent buildings. To handle these challenges, we present a new building instance mapping framework for ALS point clouds guided by polygonal building footprints. The key idea is to generate potential building instances by merging ALS points into segments hierarchically. Then, the selection of correct building instances is realized by registering building candidates and 2-D polygons. The proposed method is mainly tested in three scenes (residential regions, urban regions, and high-rise buildings) with various building architectures. The experimental results show that our framework can still achieve high instance-level mapping accuracy even if there are large positioning errors among polygons. Furthermore, compared with the existing solutions, the proposed method also performs much better in challenging scenes. These detailed experiments demonstrate that it is feasible to extract large-scale building instances with high accuracy from ALS point clouds aided by the widely accessible polygonal footprints.

Future work will address the reduction of the effects of classification errors in preprocessing and inconsistency between ALS point clouds and polygonal footprints. We have focused on building instance extraction from ALS data and have not addressed the delineation of extracted instances and updating of polygonal footprints. This also requires further study.

References

[1] D. Marmanis, K. Schindler, J. D. Wegner, S. Galliani, M. Datcu, and U. Stilla, "Classification with an edge: Improving semantic image segmentation with boundary detection," *ISPRS J. Photogramm. Remote Sens.*, vol. 135, pp. 158–172, Jan. 2018.

[2] L. Mou, Y. Hua, and X. X. Zhu, "A relation-augmented fully convolutional network for semantic segmentation in aerial scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12416–12425.

[3] Z. Zhang *et al.*, "A multilevel point-cluster-based discriminative feature for ALS point cloud classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 6, pp. 3309–3321, Jun. 2016.

[4] X. Ge, B. Wu, Y. Li, and H. Hu, "A multi-primitive-based hierarchical optimal approach for semantic labeling of ALS point clouds," *Remote Sens.*, vol. 11, no. 10, p. 1243, May 2019.

[5] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 740–755.

[6] A. Shakeel, W. Sultani, and M. Ali, "Deep built-structure counting in satellite imagery using attention based re-weighting," *ISPRS J. Photogramm. Remote Sens.*, vol. 151, pp. 313–321, May 2019.

[7] J. Kang, M. Körner, Y. Wang, H. Taubenböck, and X. X. Zhu, "Building instance classification using street view images," *ISPRS J. Photogramm. Remote Sens.*, vol. 145, pp. 44–59, Nov. 2018.

[8] M. Rutzinger, B. Höfle, S. O. Elberink, and G. Vosselman, "Feasibility of facade footprint extraction from mobile laser scanning data," *Photogrammetrie-Fernerkundung-Geoinf.*, vol. 2011, no. 3, pp. 97–107, 2011.

[9] S. Xia and R. Wang, "Semiautomatic construction of 2-D Façade footprints from mobile LiDAR data," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 6, pp. 4005–4020, Jun. 2019.

[10] M. Li, L. Nan, and S. Liu, "Fitting boxes to manhattan scenes using linear integer programming," *Int. J. Digit. Earth*, vol. 9, no. 8, pp. 806–817, Aug. 2016.

[11] J. Du *et al.*, "A novel framework for 2.5-D building contouring from large-scale residential scenes," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 6, pp. 4121–4145, Jun. 2019.

[12] S. Ji, Y. Shen, M. Lu, and Y. Zhang, "Building instance change detection from large-scale aerial images using convolutional neural networks and simulated samples," *Remote Sens.*, vol. 11, no. 11, p. 1343, Jun. 2019.

[13] B. Yu, H. Liu, J. Wu, Y. Hu, and L. Zhang, "Automated derivation of urban building density information using airborne LiDAR data and object-based method," *Landscape Urban Planning*, vol. 98, nos. 3–4, pp. 210–219, Dec. 2010.

[14] Z. Liqiang, D. Hao, C. Dong, and W. Zhen, "A spatial cognition-based urban building clustering approach and its applications," *Int. J. Geograph. Inf. Sci.*, vol. 27, no. 4, pp. 721–740, Apr. 2013.

[15] W. Zhao, Y. Bo, J. Chen, D. Tiede, T. Blaschke, and W. J. Emery, "Exploring semantic elements for urban scene recognition: Deep integration of high-resolution imagery and OpenStreetMap (OSM)," *ISPRS J. Photogramm. Remote Sens.*, vol. 151, pp. 237–250, May 2019.

[16] S. Du, Y. Zhang, Z. Zou, S. Xu, X. He, and S. Chen, "Automatic building extraction from LiDAR data fusion of point and grid-based features," *ISPRS J. Photogramm. Remote Sens.*, vol. 130, pp. 294–307, Aug. 2017.

[17] M. Soilán, B. Riveiro, P. Liñares, and A. Pérez-Rivas, "Automatic parametrization of urban areas using ALS data: The case study of santiago de compostela," *ISPRS Int. J. Geo-Inf.*, vol. 7, no. 11, p. 439, Nov. 2018. [Online]. Available: https://www.mdpi.com/2220-9964/7/11/439

[18] M. Cote and P. Saeedi, "Automatic rooftop extraction in nadir aerial imagery of suburban regions using corners and variational level set evolution," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 1, pp. 313–328, Jan. 2013.

[19] X. Wang *et al.*, "A robust segmentation framework for closely packed buildings from airborne LiDAR point clouds," *Int. J. Remote Sens.*, vol. 41, no. 14, pp. 5147–5165, Jul. 2020.

[20] M. Awrangjeb, M. Ravanbakhsh, and C. S. Fraser, "Automatic detection of residential buildings using LiDAR data and multispectral imagery," *ISPRS J. Photogramm. Remote Sens.*, vol. 65, no. 5, pp. 457–467, Sep. 2010.

[21] Y. Yu, J. Li, H. Guan, and C. Wang, "A marked point process for automated building detection from LiDAR point-clouds," *Remote Sens. Lett.*, vol. 4, no. 11, pp. 1127–1136, Nov. 2013.

[22] B. Yang, W. Xu, and Z. Dong, "Automated extraction of building outlines from airborne laser scanning point clouds," *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 6, pp. 1399–1403, Nov. 2013.

[23] S. Xia and R. Wang, "Extraction of residential building instances in suburban areas from mobile LiDAR data," *ISPRS J. Photogramm. Remote Sens.*, vol. 144, pp. 453–468, Oct. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0924271618302272

[24] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 652–660.

[25] L. Wang, Y. Huang, Y. Hou, S. Zhang, and J. Shan, "Graph attention convolution for point cloud semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10296–10305.

[26] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3D point clouds: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Jun. 29, 2020, doi: 10.1109/TPAMI.2020.3005434.

[27] B. Yang *et al.*, "Learning object bounding boxes for 3D instance segmentation on point clouds," 2019, *arXiv:1906.01140*. [Online]. Available: http://arxiv.org/abs/1906.01140

[28] L. Han, T. Zheng, L. Xu, and L. Fang, "OccuSeg: Occupancy-aware 3D instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2940–2949.

[29] L. Jiang, H. Zhao, S. Shi, S. Liu, C.-W. Fu, and J. Jia, "PointGroup: Dual-set point grouping for 3D instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4867–4876.

[30] Q. Hu *et al.*, "RandLA-Net: Efficient semantic segmentation of large-scale point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11108–11117.

[31] M. Yousefhussien, D. J. Kelbe, E. J. Ientilucci, and C. Salvaggio, "A multi-scale fully convolutional network for semantic labeling of 3D point clouds," *ISPRS J. Photogramm. Remote Sens.*, vol. 143, pp. 191–204, Sep. 2018.

[32] L. Winiwarter, G. Mandlburger, S. Schmohl, and N. Pfeifer, "Classification of ALS point clouds using end-to-end deep learning," *PFG-J. Photogramm., Remote Sens. Geoinf. Sci.*, vol. 87, no. 3, pp. 75–90, Sep. 2019.

[33] Y. Li, L. Ma, Z. Zhong, D. Cao, and J. Li, "TGNet: Geometric graph CNN on 3-D point cloud segmentation," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 5, pp. 3588–3600, May 2020.

[34] W. Li, F.-D. Wang, and G.-S. Xia, "A geometry-attentional network for ALS point cloud classification," *ISPRS J. Photogramm. Remote Sens.*, vol. 164, pp. 26–40, Jun. 2020.

[35] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2189–2202, Nov. 2012.

[36] H. Fan, W. Yao, and Q. Fu, "Segmentation of sloped roofs from airborne LiDAR point clouds using ridge-based hierarchical decomposition," *Remote Sens.*, vol. 6, no. 4, pp. 3284–3301, Apr. 2014.

[37] H. Fan, A. Zipf, Q. Fu, and P. Neis, "Quality assessment for building footprints data on OpenStreetMap," *Int. J. Geograph. Inf. Sci.*, vol. 28, no. 4, pp. 700–719, Apr. 2014.

[38] H. Du, N. Alechina, M. Jackson, and G. Hart, "A method for matching crowd-sourced and authoritative geospatial data," *Trans. GIS*, vol. 21, no. 2, pp. 406–427, Apr. 2017.

[39] X. Tong, D. Liang, and Y. Jin, "A linear road object matching method for conflation based on optimization and logistic regression," *Int. J. Geograph. Inf. Sci.*, vol. 28, no. 4, pp. 824–846, Apr. 2014.

[40] H. Fan, B. Yang, A. Zipf, and A. Rousell, "A polygon-based approach for matching OpenStreetMap road networks with regional transit authority data," *Int. J. Geograph. Inf. Sci.*, vol. 30, no. 4, pp. 748–764, Apr. 2016.

[41] A. Chehreghan and R. A. Abbaspour, "A geometric-based approach for road matching on multi-scale datasets using a genetic algorithm," *Cartogr. Geograph. Inf. Sci.*, vol. 45, no. 3, pp. 255–269, May 2018.

[42] K. Khoshelham and B. Gorte, "Registering pointclouds of polyhedral buildings to 2D maps," in *Proc. 3rd ISPRS Int. Workshop 3D-ARCH, 3D Virtual Reconstruction Vis. Complex Archit.*, Trento, Italy, Feb. 2009, pp. 1–7.

[43] B. Vallet, M. Pierrot-Deseilligny, D. Boldo, and M. Brédif, "Building footprint database improvement for 3D reconstruction: A split and merge approach and its evaluation," *ISPRS J. Photogramm. Remote Sens.*, vol. 66, no. 5, pp. 732–742, Sep. 2011.

[44] X. Zhang, W. Yin, M. Yang, T. Ai, and J. Stoter, "Updating authoritative spatial data from timely sources: A multiple representation approach," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 72, pp. 42–56, Oct. 2018.

[45] X. Zhuo, F. Fraundorfer, F. Kurz, and P. Reinartz, "Optimization of OpenStreetMap building footprints based on semantic information of oblique UAV images," *Remote Sens.*, vol. 10, no. 4, p. 624, Apr. 2018.

[46] P. Kaiser, J. D. Wegner, A. Lucchi, M. Jaggi, T. Hofmann, and K. Schindler, "Learning aerial image segmentation from online maps," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 11, pp. 6054–6068, Nov. 2017.

[47] J. Huang, X. Zhang, Q. Xin, Y. Sun, and P. Zhang, "Automatic building extraction from high-resolution aerial images and LiDAR data using gated residual refinement network," *ISPRS J. Photogramm. Remote Sens.*, vol. 151, pp. 91–105, May 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0924271619300590

[48] N. Girard, G. Charpiat, and Y. Tarabalka, "Aligning and updating cadaster maps with aerial images by multi-task, multi-resolution deep learning," in *Proc. Asian Conf. Comput. Vis.* Cham, Switzerland: Springer, 2018, pp. 675–690.

[49] A. Zampieri, G. Charpiat, N. Girard, and Y. Tarabalka, "Multimodal image alignment through a multiscale chain of neural networks with application to remote sensing," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 657–673.

[50] H. Chen, W. Xie, A. Vedaldi, and A. Zisserman, "AutoCorrect: Deep inductive alignment of noisy geometric annotations," 2019, *arXiv:1908.05263*. [Online]. Available: http://arxiv.org/abs/1908.05263

[51] Y. Zhang, W. Yang, X. Liu, Y. Wan, X. Zhu, and Y. Tan, "Unsupervised building instance segmentation of airborne LiDAR point clouds for parallel reconstruction analysis," *Remote Sens.*, vol. 13, no. 6, p. 1136, Mar. 2021.

[52] L. Xu, C. Lu, Y. Xu, and J. Jia, "Image smoothing via L_0 gradient minimization," *ACM Trans. Graph.*, vol. 30, no. 6, p. 174, 2011.

[53] F. Lafarge and C. Mallet, "Creating large-scale city models from 3D-point clouds: A robust approach with hybrid representation," *Int. J. Comput. Vis.*, vol. 99, no. 1, pp. 69–85, Aug. 2012, doi: 10.1007/s11263-012-0517-8.

[54] W. Zhang *et al.*, "An easy-to-use airborne LiDAR data filtering method based on cloth simulation," *Remote Sens.*, vol. 8, no. 6, p. 501, Jun. 2016. [Online]. Available: https://www.mdpi.com/2072-4292/8/6/501

[55] A. Sampath and J. Shan, "Segmentation and reconstruction of polyhedral building roofs from aerial LiDAR point clouds," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 3, pp. 1554–1567, Mar. 2010.

[56] J. Yan, J. Shan, and W. Jiang, "A global optimization approach to roof segmentation from airborne LiDAR point clouds," *ISPRS J. Photogramm. Remote Sens.*, vol. 94, pp. 183–193, Aug. 2014.

[57] Y. Xu, W. Yao, L. Hoegner, and U. Stilla, "Segmentation of building roofs from airborne LiDAR point clouds using robust voxel-based region growing," *Remote Sens. Lett.*, vol. 8, no. 11, pp. 1062–1071, Nov. 2017.

[58] Y. Sun, S. Schaefer, and W. Wang, "Denoising point sets via l0 minimization," *Comput. Aided Geometric Des.*, vols. 35–36, pp. 2–15, May 2015.

[59] Q. Yan, L. Xu, J. Shi, and J. Jia, "Hierarchical saliency detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 1155–1162.

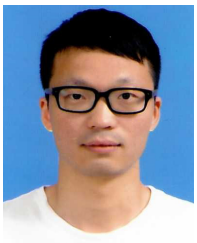
[60] C. Godsil and G. F. Royle, *Algebraic Graph Theory*, vol. 207. Berlin, Germany: Springer, 2001.

[61] Gurobi Optimization. (2019). *Gurobi Optimizer Reference Manual*. [Online]. Available: http://www.gurobi.com

[62] D. Chen, L. Zhang, P. T. Mathiopoulos, and X. Huang, "A methodology for automated segmentation and reconstruction of urban 3-D buildings from ALS point clouds," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 10, pp. 4199–4217, Oct. 2014.

Shaobo Xia received the bachelor's degree in geodesy and geomatics from the School of Geodesy and Geomatics, Wuhan University, Wuhan, China, in 2013, the master's degree in cartography and geographic information systems from the University of Chinese Academy of Sciences, Beijing, China, in 2016, and the Ph.D. degree in geomatics from the University of Calgary, Calgary, AB, Canada, in 2020.

He is an Assistant Professor with the College of Environment and Resources, Zhejiang A&F University, Hangzhou, China. His research interests include point cloud processing and urban mapping.



Sheng Xu received the B.Eng. degree in computer science and technology from Nanjing Forestry University, Nanjing, China, in 2010, and the Ph.D. degree in digital image systems from the University of Calgary, Calgary, AB, Canada, in 2018.

In 2018, he joined the College of Information Science and Technology, Nanjing Forestry University, where he is an Associate Professor. His research interests include mobile mapping, vegetation mapping, and computer vision.

Ruisheng Wang (Senior Member, IEEE) received the B.Eng. degree from Wuhan University, Wuhan, China, in 1993, the M.Sc.E. degree from the University of New Brunswick, Fredericton, NB, Canada, in 2004, and the Ph.D. degree from McGill University, Montreal, QC, Canada, in 2011.

He is an Associate Professor with the Department of Geomatics Engineering, University of Calgary, Calgary, AB, Canada. His research interests include geomatics and computer vision, especially point cloud processing.

Jonathan Li (Senior Member, IEEE) received the Ph.D. degree in remote sensing and geographic information system (GIS) from the University of Cape Town, Cape Town, South Africa, in 2000.

He is a Professor of geomatics and the Head of the Geospatial Technology and Remote Sensing (GeoSTARS) Group, Department of Geography & Environmental Management, University of Waterloo, Waterloo, ON, Canada. He is also an Adjunct Professor with York University, Toronto, ON, Canada, and a guest professor of several top-ranked universities in China. He has published extensively in leading remote sensing journals. His research is focused on the use of mobile laser scanning point clouds for 3-D surface modeling.

Dr. Li was a recipient of several prestigious awards. He was the Chair of the International Society for Photogrammetry and Remote Sensing (ISPRS) Inter Commission Working Group (ICWG) I/Va on Mobile Scanning and Imaging Systems from 2012 to 2016 and the FIG WG4.4 on Maritime and Marine Spatial Information Management from 2014 to 2018 and the Vice-Chair of the ICA Commission on Mapping from Remote Sensor Imagery from 2011 to 2015 and the FIG Commission IV on Hydrography. He has been the Remote Sensing Editor of *Geomatica* since 2007.

Guanghui Wang (Senior Member, IEEE) was with the University of Kansas, Lawrence, KS, USA, where he was an Assistant Professor and an Associate Professor from 2014 to 2020. He is an Associate Professor with Ryerson University, Toronto, ON, Canada. He has authored one book *Guide to Three Dimensional Structure and Motion Factorization* (Springer-Verlag). He has published over 130 articles in peer-reviewed journals and conferences. His research interests include computer vision, image analysis, machine learning, and intelligent systems.

Dr. Wang has served as associate editor and on the editorial board of 2 journals, the area chair or a TPC member of more than 30 conferences, and a reviewer of more than 30 journals.