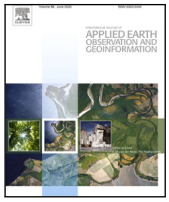
International Journal of Applied Earth Observations and Geoinformation 114 (2022) 103042

Contents lists available at ScienceDirect

International Journal of Applied Earth Observation and Geoinformation

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



Automated extraction of building instances from dual-channel airborne LiDAR point clouds

Huifang Feng [a], Yiping Chen [b,*], Zhipeng Luo [a], Wentao Sun [a], Wen Li [a], Jonathan Li [c,*]

[a] Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Informatics, Xiamen University, Xiamen 361005, China
[b] School of Geospatial Engineering and Science, Sun Yat-sen University, 519082 Zhuhai, China
[c] Department of Geography and Environmental Management and Department of Systems Design Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada

ARTICLE INFO

Keywords:
Dual-channel airborne LiDAR
Reorganization
Preprocessing-free
Building instance extraction
Point clouds

ABSTRACT

With the rapid development of Light Detection And Ranging (LiDAR) systems, the novel dual-channel airborne LiDAR systems have emerged to provide more complete and precise data than traditional scanners for building instance extraction since 2013. RIEGL VQ-1560i, launched in 2016, is a state-of-the-art dual-channel LiDAR system, which is capable of capturing dense points on building rooftops and façades simultaneously, due to the unique and innovative bidirectional scanning angle. Our proposed method is the first ever to use dual-channel airborne LiDAR data for subsequent point clouds processing. The main challenges of the new LiDAR data are significant amount of points, complex data structure and multi-class targets. We proposed a preprocessing-free building instance extraction method consisting of three steps, i.e., point cloud reorganization, rasterization, and constraint-based labeling for improving the extraction performance. First, point cloud reorganization, consisting of point distribution-based slicing, coarse 3D semantic segmentation, and top-down merging, is used to reorganize point cloud scene into interrelated point groups. This greatly reduces the processing difficulty and computational burden of complex structures while removing multiple classes of non-building points. Second, we rasterize the point groups into images to further reduce computational complexity while improving processing efficiency. Finally, we utilize the upper and lower structural relationship of buildings to label them and then remap into 3D buildings. Experimental results on six test point cloud scenes demonstrate the outstanding performance of the proposed preprocessing-free method. For semantic-level performance, our method achieves 95.36% in average recall and 93.59% in average F1-score. While for instance-level performance, our approach reaches 92.86% and 98.31% in quality on two public test scenes, respectively.

1. Introduction

LiDAR scanning is an active data collection that is less affected by weather conditions than optical imaging. The rapid development of LiDAR technology has provided the surveying and mapping industry with better and more accurate data. RIEGL VQ-1560i is a state-of-the-art LiDAR system consisting of two LiDAR channels of same laser wavelength, a high-grade Applanix AP60 IMU/GNSS system, a PhaseOne 100 MPix camera, and an optional integrated camera (RIEGL, 2019), as shown in Fig. 1. This dual-channel system can operate at an altitude of up to 5600 m and has a maximum measurement speed of up to 1.33 million per second (meas./s). The specifications of RIEGL VQ-1560i is detailed in Table 1. Furthermore, the dual-channel design provides a unique revolutionary front/rear bidirectional simultaneous scanning function, enabling VQ1560i to more effectively and accurately capture high point density data from multiple angles within 60 degrees Field of View (FoV), the data acquisition illustration is shown in Fig. 2. Due to its Multiple-Time-Around (MTA) processing of up to 20 pulses simultaneously in the air, RIEGL VQ-1560i can acquire point clouds of façades between buildings with angles less than 16 degrees. Hence, RIEGL VQ-1560i can provide the rooftops and façades point clouds of buildings simultaneously.

Automated extraction of building instances from point clouds is playing an increasingly important role in many applications, including disaster management (Dash et al., 2004), urban analysis (Yu et al., 2010), cadastral surveying (Rutzinger et al., 2011), change detection (Qin et al., 2015), urban context-aware visualization (Deng et al., 2016), and 3D reconstruction (Chen et al., 2017). Traditionally, 3D building extraction has basically used LiDAR data from various laser scanning systems. Single-channel Airborne Laser Scanning (ALS) systems mainly collect points on building rooftops but hardly cover the

* Corresponding authors.
E-mail addresses: chenyiping@xmu.edu.cn (Y. Chen), junli@uwaterloo.ca (J. Li).

https://doi.org/10.1016/j.jag.2022.103042
Received 18 May 2022; Received in revised form 23 September 2022; Accepted 28 September 2022
Available online 17 October 2022

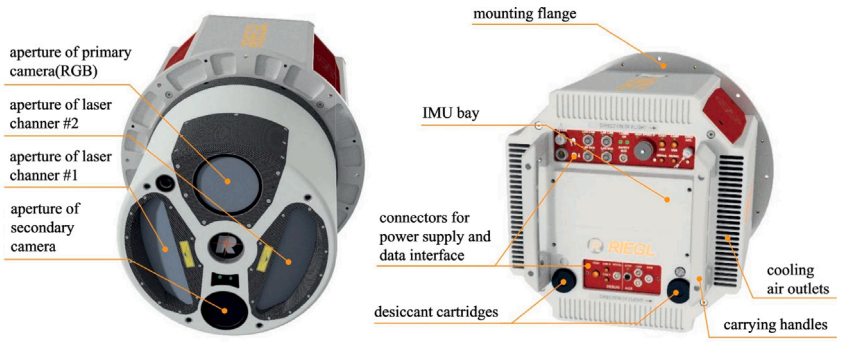


Fig. 1. Architecture of RIEGL VQ-1560i.

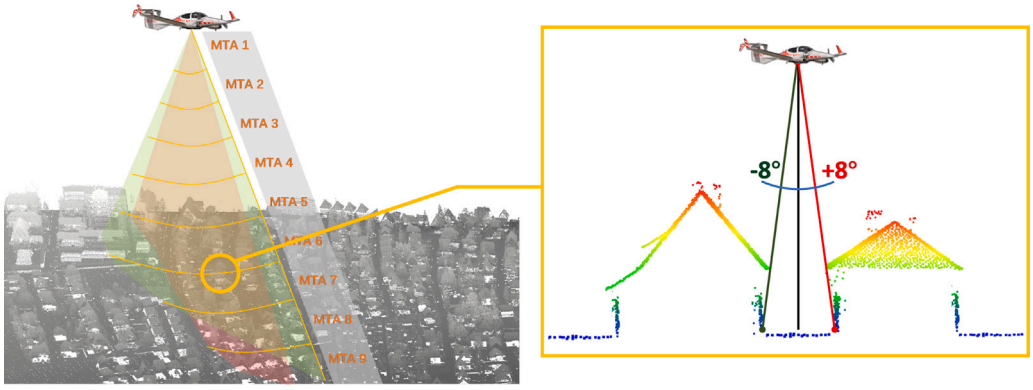


Fig. 2. Data acquisition of RIEGL VQ-1560i.

Table 1
Specifications of the RIEGL VQ-1560i.

Average point density (pts/m^2)	2	8	20	60
Flight altitude (m)	2000	1370	1000	351
Ground speed (kn)	300	210	115	110
Swath width (m)	2240	1540	1130	400
Productivity (km^2/h)	996	480	192	64
Measurement rate (meas./s)	933,000	1.33 million	1.33 million	1.33 million
Camera ground sampling distance (mm)	114	103	75	26
Camera trigger interval (s)	3.5	4.1	5.4	2.0

façades. Terrestrial Laser Scanning (TLS) systems only provide data on façades, and TLS is difficult to construct large-scale and complex terrestrial scans due to the low scanning efficiency. Mobile Laser Scanning (MLS) systems can acquire detailed data in large-scale urban scenes other than rooftops. As a result, the traditional LiDAR data-based 3D building extraction requires separate processing or pre-registration of rooftop and façade point clouds. However, due to information loss and inaccurate registration, the extracted building instances suffer from low completeness and correctness, which is especially obvious in complex large-scale urban scenes. Compared with other LiDAR systems, the dual-channel system can provide denser and more accurate rooftop and facade point clouds of buildings simultaneously, significantly increasing the collection of ultra-wide scenes and complex urban environments (Xie et al., 2020). The comparison of the dual-channel ALS system and other systems is shown in Table 2.

For a point cloud scene, the point density on the ground is usually higher than elsewhere because the ground is continuous, flat, and impenetrable to laser. The non-ground objects have a vertical geometry, but their point distribution is disorganized and uneven due to different shapes, reflectivity, and scan angles. In addition, the complex and incomplete structure of buildings in the real-world, with occlusions and local similarities between different buildings. Moreover, the laser scanning process generates sparse outliers due to measurement errors, which complicates the point cloud local features and leads to processing errors. Compared with other LiDAR data, the dual-channel LiDAR data provides denser and more detailed point cloud scenes, which presents more stringent requirements and challenges for 3D building extraction. (1) How to handle the expensive computation burden raised by the significant amount of point clouds. (2) How to extract complete building from point cloud scenes with complex data structure. (3) How to avoid the interference of multi-class targets in the point cloud scenes.

Motivated by the above limitations and the emerging processing requirements of dual-channel LiDAR data, we propose a new approach to automated extraction of complete buildings, including the rooftops and façades, from large-scale dual-channel airborne LiDAR point cloud scenes. We list the contributions of this paper below.

(1) This paper is the first ever to use dual-channel airborne LiDAR data for building instance extraction that benefits from the rooftops and façades point clouds.

(2) This paper presents the first preprocessing-free building instance extraction method for large-scale point cloud scenes, which improves the automation of the algorithm for extracting buildings from point cloud scenes.

(3) We propose a novel point cloud scene reorganization method, utilizing a point distribution and top-down merging strategy, which significantly weakens the influence of outliers and noise in point cloud scenes and reduces computational complexity of subsequent operations.

Table 2
Comparison of different LiDAR systems.

Property	TLS	MLS	ALS	Dual-channel ALS
Density	>100 pts/m^2	>100 pts/m^2	<20 pts/m^2	<60 pts/m^2
Accuracy	mm-level	mm-level	<15 cm	<20 mm
Area	Specific small areas	Street areas	Wide areas; urban environments	Ultra-wide areas; complex urban environments
Data	Static objects; vegetation; building façades	Road objects; vegetation; building façades	Remote sensing objects; building rooftops and sparse façades	Remote sensing objects; building rooftops and denser façades
Application	Small-area 3D reconstruction	HD map; urban monitoring	City modeling; urban monitoring; vegetation monitoring; etc.	City modeling; urban monitoring; vegetation monitoring; etc.

The rest of this paper is organized as follows. In Section 2, we briefly review the building extraction methods for point clouds. In Section 3, we describe the proposed preprocessing-free building instance extraction method. The experimental results on six test scenes are presented and discussed in Section 4. And Section 5 concludes the paper.

2. Related work

With the development of LiDAR technology, various LiDAR systems emerged to provide different LiDAR data. Recently, researchers have made important progress in building extraction from different LiDAR data, which can be categorized into TLS/MLS data-based methods and ALS data-based methods.

TLS/MLS data-based methods. Since TLS/MLS data provides detailed information about the building façades while can hardly cover the rooftops, TLS/MLS data-based building extraction methods generally focus on the extraction of façades. Pu and Vosselman (2009) proposed a knowledge rules-based building façade recognition method, but it can hardly be applied to urban scenes (Wang et al., 2020). Yang et al. (2013a) proposed a projection-based method for 2D building outlines extraction, but it suffers from heavy computational burden (Che et al., 2019). Some methods (Gao and Yang, 2013; Fan et al., 2014) utilized the spatial distribution patterns to extract buildings, but their performance was limited by the data quality (Wang et al., 2020). Xia and Wang (2018) proposed an objected-oriented method to locate and segment building instances utilize the clues of façades, but the building components identification is restricted by the scene segmentation. A semantic-based method was proposed by Wang et al. (2020), utilizing a Markov Random Field optimization model. However, the extraction completeness was limited by variable point densities and data gaps.

ALS data-based methods. Due to the sparseness of ALS data, which only provides incomplete buildings with rooftops and sparse façades, some ALS data-based building extraction methods take advantage of Digital Surface Model (DSM) and multi-sensor data. Du et al. (2017) proposed a feature-retained DSM interpolation method to extract buildings, but it suffered from interpolation error and information loss. Awrangjeb et al. (2013) extracted 3D roof from point clouds and multispectral images, but it is limited by the extracted 2D features from the orthoimage (Awrangjeb and Fraser, 2014). Huang et al. (2018) proposed a method to remove non-building points and extract buildings using topological, geometric and penetrating features, but it cannot detect low-density or small-area buildings. Nguyen et al. (2020) proposed a snake model-based method, which use combination information from point clouds and optical image to extract buildings, but only the height information of point clouds plays a role during the processing (Hui et al., 2021). Zhang et al. (2020) extracted buildings utilizing features from remote sensing images and point clouds, but the required data fusion not only leads to a decrease in the reliability of building extraction, but also increases the computational burden.

Some other ALS data-based methods focus on extracting buildings from the pre-segmented non-ground points. Yang et al. (2013b) utilized the point marking for building outlines extraction from non-ground points, but it cannot separate spatially connected buildings. Awrangjeb and Fraser (2014) constructed a building mask and clustered similar surfaces based on coplanarity feature to extract building roofs from non-ground points, but the incomplete vegetation removal results in low boundary accuracy. Albers et al. (2016) proposed a Hough transform-based method for building extraction and an energy minimization approach for regularization, but it is limited by the building orientation and peak detection. Cai et al. (2019) utilized fuzzy C-means and region growing to extract buildings, but the performance depends on parameter setting and the geometrical properties (Meng et al., 2009). Liu et al. (2020) proposed a building extraction method based on minimum cut and improved post-processing. Hui et al. (2021) proposed a multi-constraint graph segmentation-based method for object-based building extraction. However, these methods require multiple pre- or post-processing to decrease computational cost and improve the extraction accuracy.

In recent, deep learning frameworks have proven to be effective in many LiDAR data-based application scenarios. Zhou and Gong (2018) made the first attempt to combine deep learning networks with ALS point cloud-based building detection, but the required filter size varies with the size of the pattern to be captured. Maltezos et al. (2019) employed a Convolutional Neural Network (CNN) framework to classify building structures from complex urban areas, but suffers from information loss during the feature extraction (Li et al., 2020). Some relevant methods (Huang et al., 2019; Yuan et al., 2021) used the fusion data of point clouds and remote sensing images as training data for deep learning networks. However, they rely on data fusion, and the training data acquisition is labor-intensive, time-consuming, and costly.

The advent of novel dual-channel ALS systems provides complete buildings with rooftops and facades. It is urgently necessary to develop new low-data-dependency building extraction methods, to achieve complete building extraction from complex dual-channel ALS data.

3. Method

This paper presents a novel instance-level building extraction method for point cloud scenes collected by a dual-channel airborne LiDAR system. The input of the proposed method is a raw point cloud scene, denoted as a point set with three-dimensional coordinates $C = (X, Y, Z)$. The output is a set of 3D building instances $B = \{B_1, B_2, \ldots, B_N\}$, where N represents the number of buildings, B_i ($i \in [1, N]$) denotes the point clouds of a building instance. The proposed method consists of three modules. (1) Point cloud reorganization. In this module, the input raw point cloud scene is split into point cloud slices, which are then segmented and merged into interrelated point groups. (2) Rasterization and detection. In this module, the interrelated point groups are rasterized into images, and based on which to detect candidate 2D buildings. (3) Labeling and extraction. In this module, the candidate 2D buildings are labeled to extract 3D building instances. The workflow of our preprocessing-free method is illustrated in Fig. 3.

3.1. Point cloud reorganization

First of all, we propose a Point Distribution-Based Slicing (PDBS) method to split the raw point cloud scene into slices. Then, we propose a Coarse 3D Semantic Segmentation (C3DSS) method to segment building points from each slice. Finally, these slices are merged into point groups based on a top-down strategy.

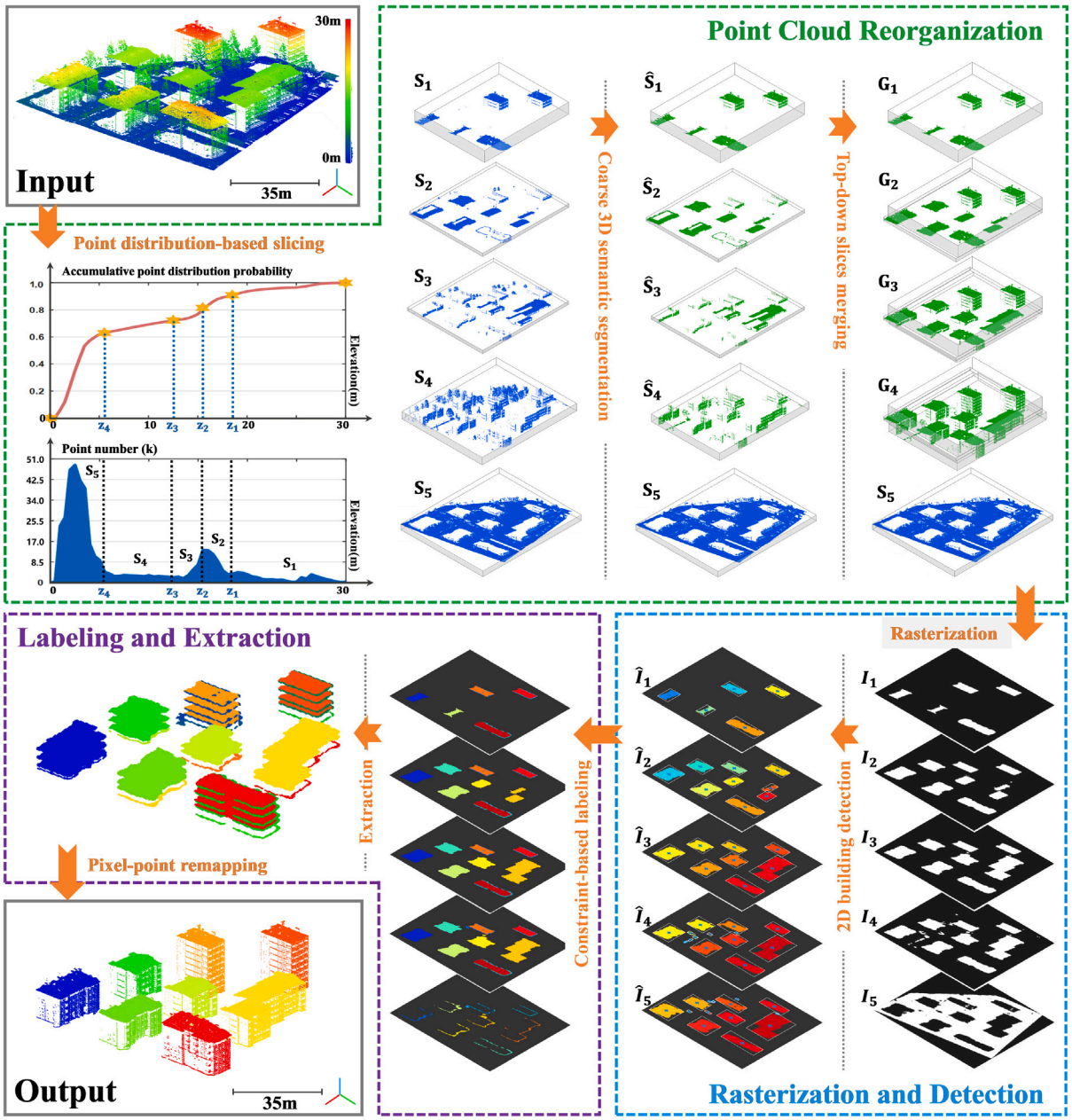


Fig. 3. Workflow of the proposed preprocessing-free building extraction method, consists of three modules: (1) point cloud reorganization module (green dashed area), (2) rasterization and detection module (blue dashed area), and (3) labeling and extraction module (purple dashed area). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3.1.1. *Point distribution-based slicing*

The proposed PDBS method splits the raw point cloud scene C into $S = \{S_1, S_2, \ldots, S_n\}$, which is detailed below.

(1) Define a unit height of C as h_u and calculate the number of unit height N_{h_u} as follows:

$$N_{h_u} = round\left(\frac{\lceil f_{max}(Z) - f_{min}(Z)\rceil}{h_u}\right) \tag{1}$$

where $round(\cdot)$ calculates the nearest integer of input value, $\lceil\cdot\rceil$ rounds input value upwards to the next integer, $f_{max}(\cdot)$ and $f_{min}(\cdot)$ calculate the maximum and minimum value, respectively.

(2) Define a point number vector C_{h_u}, a probability vector P_{h_u}, and $C_{h_u}(i)$ is the number of points that located in the i_{th} unit height in C. For $C_{h_u}(i)$, its corresponding point distribution probability $P_{h_u}(i)$ is calculated as follows:

$$P_{h_u}(i) = \frac{num\left(C_{h_u}(i)\right)}{num(C)}, \quad i \in \left[1, N_{h_u}\right] \tag{2}$$

where $num(\cdot)$ counts the points number.

(3) Define a cumulative probability vector $\hat{P}_{h_u}$, for $\hat{P}(i)_{h_u} \in \hat{P}_{h_u}$, calculated as follows:

$$\hat{P}_{h_u}(i) = \cup_1^i P_{h_u}(i), \quad i \in \left[1, N_{h_u}\right] \tag{3}$$

(4) Define a candidate probability set P_{can}, calculated as follows:

$$P_{can}(j) = \begin{cases} \hat{P}_{h_u}(j), & \frac{\sum_{j-sd}^{j-1} P_{h_u}(k)}{\sum_{j}^{j+sd\cdot sc} P_{h_u}(k)} > 2, j \in \left[sd+1, \left\lfloor \frac{N_{h_u}}{2} \right\rfloor\right] \\ 0, & otherwise \end{cases} \tag{4}$$

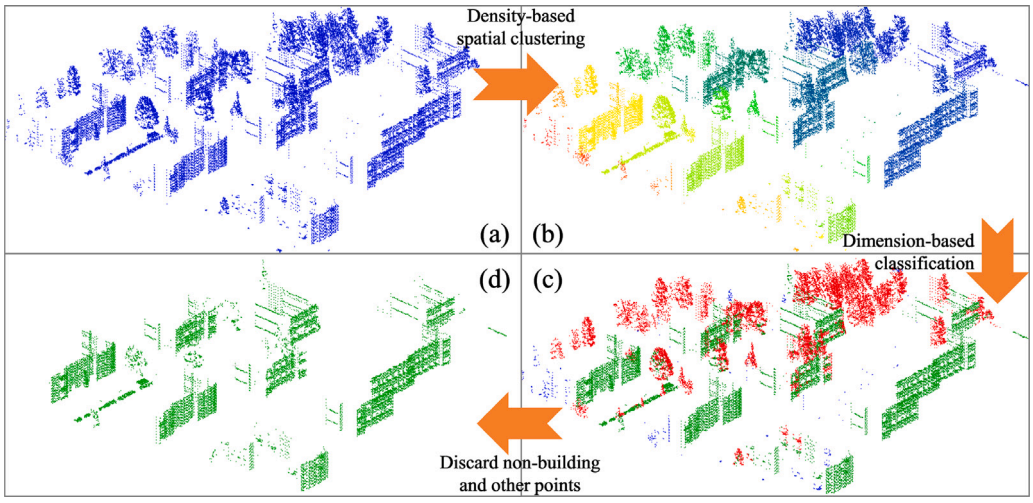


Fig. 4. Illustration of the proposed C3DSS method. (a) Slice obtained through PDBS. (b) Density-based spatial clustering results, with different colors representing different clusters. (c) Dimension-based classification results. Green, red, and blue represent building, non-building, and other points. (d) Building slice. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where $\lfloor\cdot\rfloor$ rounds input value down to the previous integer, sd and sc are constants, and sd denotes the stride, $sd < \frac{N_{h_u}}{2} - 1$, and sc denotes the repeat number.

(5) Define the point probability of S_n as $P_{sli}(n)$, calculated as follows:

$$P_{sli}(n) = f_{max}\left(P_{can}\right) \tag{5}$$

(6) Assuming that the maximum height of slice S_n located in the k_{th} unit height, then the maximum height of S_n equals to $k \times h_u$. Denote the k_{th} value of $\hat{P}_{h_u}$ as $\hat{P}_{h_u}(k)$, which is almost equals to $P_{sli}(n)$. Based on which to obtain the value of k. And then, a set of value Z_{spl} can be calculated as follows:

$$Z_{spl}(i) = \begin{cases} f_{max}(Z) - i \cdot \frac{f_{max}(Z) - Z_{spl}(n-1)}{n-1}, & 1 \le i < n-1 \\ k \cdot h_u, & i = n-1 \end{cases} \tag{6}$$

(7) C is segmented into n slices using $n-1$ horizontal planes, each determined by a value in Z_{spl}. The slices are referred from top to bottom as to $\{S_1, \ldots, S_n\}$. In the following processing, S_n is called the basis slice, and $\{S_1, \ldots, S_{n-1}\}$ the target slices.

The proposed PDBS method has significant advantages compared to slicing at constant intervals along the z-axis. (1) PDBS can divide almost all (but not only) ground points into the basis slice according to iterative point distribution calculation, avoiding ground filtering preprocessing. (2) PDBS dynamically determines the height of each target slice. Thinner slices in dense regions reduce structural complexity, mutual interference of objects, and computational burden. Slices of sparse regions are thicker to preserve local structural features, helping to obtain more complete building segmentation results. (3) PDBS is robust and its results are not disturbed by outliers or noise.

3.1.2. Coarse 3D semantic segmentation

We proposed the C3DSS method to segment building points from each target slice, the workflow of C3DSS is shown in Fig. 4. Firstly, the target slice (Fig. 4(a)) was separated into different clusters (Fig. 4(b)) through a density-based spatial clustering. Secondly, each cluster was classified into building, non-building, and other points (Fig. 4(c)) according to the dimension-based features in the cluster. Finally, discard all the non-building and other points to obtain the building slice (Fig. 4(d)).

(1) Density-based spatial clustering. Define the minimum coverage rectangle of $C = (X, Y, Z)$ on xy-plane as Eq. (7). The raster scalar Θ_{scalar} is defined according to the geometric features of C as follows:

$$Rect(C) = \left[f_{min}(X),\ f_{min}(Y),\ f_{max}(X),\ f_{max}(Y)\right] \tag{7}$$

$$Area(Rect(C)) = Range(X) \cdot Range(Y) \tag{8}$$

$$\Theta_{scalar} = \sqrt{\frac{num(C)}{\sigma_{num}}} + \sqrt{\frac{Area(Rect(C))}{\sigma_{area}}} \tag{9}$$

where $Range(\cdot)$ calculates the value of $f_{max}(\cdot) - f_{min}(\cdot)$, σ_{num} and σ_{area} are constants related to point number and area of the point cloud scene, respectively. Then, the clustering method DBSCAN (Ester et al., 1996) is performed for each target slice, ensuring that the distance between any points in any two clusters is not less than $D_{min} = 2\Theta_{scalar}$.

(2) Dimension-based hierarchical classification. Firstly, define a hierarchical threshold $\theta_{num}(i)$, $i \in [1, n-1]$ as follows:

$$\theta_{num}(i) = round\left(log\left(Area\left(Rect\left(S_i\right)\right) \cdot Range\left(Z_{S_i}\right)\right)\right) \tag{10}$$

Secondly, for a cluster $c_{ij} \in S_i$, $i \in [1, n-1]$, $\forall q_j \in c_{ij}$, we use the k-Nearest Neighbors (k-NN) to search for the $\theta_{num}(i)$ neighbors in c_{ij} for q_j, denotes as $\mathcal{N}(q_j) \subset c_{ij}$. The eigenvalues λ_1, λ_2, and λ_3 $(\lambda_1 \ge \lambda_2 \ge \lambda_3)$ are generated through principal component analysis, and the 3D features can be calculated as follows:

$$\left(a_{1D},\ a_{2D},\ a_{3D}\right) = \left(\frac{\lambda_1 - \lambda_2}{\lambda_1},\ \frac{\lambda_2 - \lambda_3}{\lambda_1},\ \frac{\lambda_3}{\lambda_1}\right) \tag{11}$$

where $a_{1D} + a_{2D} + a_{3D} = 1$.

Through observing, when $a_{2D} > a_{3D}$, q_j is often in a plane structure, such as rooftop and façade, and we classify q_j to a building point. While for a stereoscopic structure, such as vegetation point, has $a_{2D} < a_{3D}$, and q_j is classified into a non-building point. Otherwise, the other points.

Finally, we define a approximate rate of cluster c_{ij} as follows:

$$building\ rate = \frac{num(building\ points)}{num\left(c_{ij}\right)} \tag{12}$$

$$non\text{-}building\ rate = \frac{num(non\text{-}building\ points)}{num\left(c_{ij}\right)} \tag{13}$$

For the cluster c_{ij}, with a given threshold θ_{rate}, all points in c_{ij} will be classified as building points if $building\ rate > \theta_{rate}$, while all points in

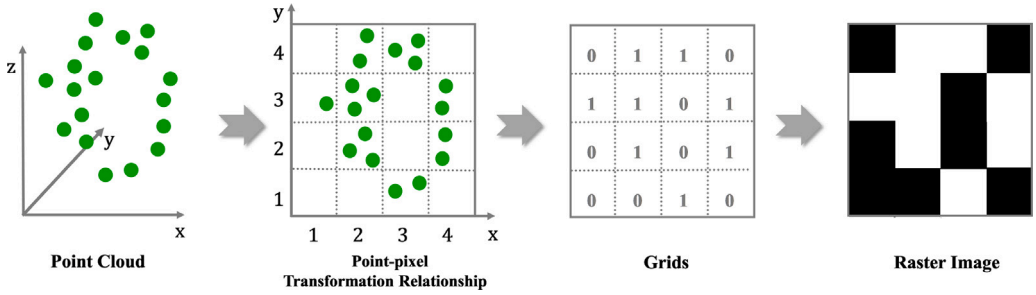


Fig. 5. Description of the rasterization details from point cloud to raster image and the point-pixel transformation relationship.

e_{ij} will be classified as non-building points if $non\text{-}building\ rate > \theta_{rate}$. Based on the classification results, discard all non-building points and obtain the candidate building slices $\{\hat{S}_1, \ldots, \hat{S}_{n-1}\}$.

3.1.3. Top-down slices merging

To enhance the structural relevance of non-ground objects among nearby slices, $\{\hat{S}_1, \ldots, \hat{S}_{n-1}\}$ are merged into point groups $G = \{G_1, \ldots, G_{n-1}\}$ in a top-down manner as follows:

$$G_i = \cup_1^i \hat{S}_i, \quad i \in [1, n-1] \tag{14}$$

Obviously, among the point groups exist internal relationships related to their sequence order, that is $G_i \subset G_j$ $(i < j)$. For a building instance, its integrity gradually increases from the rooftop to the façade as the number of slices increases during the merging. The contextual association retained in the point groups, including the structural connection sequence and location relationships of building segments, helps achieve complete extraction of building instances.

3.2. Rasterization and detection

This module consists of a rasterization operation and a proposed context-based 2D detection method. The first operation rasterizes the point groups G and the basis slice S_n into raster images $I = \{I_1, \ldots, I_n\}$. Geometric building features are utilized in context-based 2D detection to find 2D candidate buildings in image I.

3.2.1. Rasterization

First, define the rasterization basis as $G_{base} = Rect(C)$, based on which all point groups and S_n are rasterized to raster images. G_{base} ensures that the corresponding raster images of G and S_n have the same resolution. The resolution of each raster image is defined as $f_r(X) \times f_r(Y)$ with a constant r_0, and $f_r(\cdot)$ is defined as follows:

$$f_r(\cdot) = round\left(\frac{r_0 \cdot \Theta_{scalar} \cdot Range(\cdot)}{f_{max}([Range(X),\ Range(Y)])}\right) \tag{15}$$

Then, for each point set, we project points satisfying the same conditions onto the same grid, with each grid representing one pixel in the corresponding raster image. If a grid is empty, assign its pixel value to 0 and record the pixel as invalid; otherwise, assign the pixel value to 1 and record it as a valid pixel. Take G_i $(i \in [1, n-1])$ as an example, I_i is the corresponding raster image of G_i. For point $q_{g_i} \in G_i$, $q_{g_i} = (x, y, z)$, the pixel position of q_{g_i} in I_i is (X_i, Y_i), which can be calculated according to Eqs. (16) and (17).

$$X_i = \frac{\lfloor (x - f_{min}(X)) \cdot (f_r(X) - 1) \rfloor}{Range(X)} + 1 \tag{16}$$

$$Y_i = \frac{\lfloor (y - f_{min}(Y)) \cdot (f_r(Y) - 1) \rfloor}{Range(Y)} + 1 \tag{17}$$

Perform the above operations on $\{G_1, \ldots, G_{n-1}\}$ and S_n to obtain a set of raster images $I = \{I_1, \ldots, I_n\}$. I_n corresponds to S_n and looks like a large polygon with multiple internal holes. The largest polygon in I_n corresponds to the largest land-covered area in G_n. The raster images $\{I_1, \ldots, I_{n-1}\}$, corresponding to $\{G_1, \ldots, G_{n-1}\}$, which all appear to be a combination of several separated solid polygons of different shapes and sizes. Each inner hole in I_n and each solid polygon in I_i corresponds to a non-ground object.

We preserve the transformation relationship between each point and its corresponding pixel, on the basis of which each valid pixel can be reduced to one or more points. To better explain the point-to-pixel transformation relationship, we illustrate it with a simple example, which is shown in Fig. 5.

3.2.2. 2D building detection

We propose a hierarchical 2D building detection method to detect 2D building boundaries on raster images. First, we use the classical filtering method (Lim, 1990) to reduce the noise in each raster image. Then, we perform the modified Moore-Neighbor tracing method (Gonzalez et al., 2004) to detect 2D building boundaries. In particular, for I_n, we perform an erosion operation before detection to enlarge the coverage area of each inner hole in I_n. We detect the boundaries of each inner hole in I_n and discard the 2D boundary corresponds to the largest land-covered area of G_n. For I_i, $i \in [1, n)$, we detect the 2D boundary of each solid polygon. Next, we define the area of a 2D boundary as the total number of pixels within and on the boundary. We drop those boundaries whose area is smaller than a given hierarchical threshold θ_i. The value of θ_i increases with i and can be calculated according to Eqs. (18), (19), and (20):

$$\theta_{dist} = \frac{4\sqrt{\Theta_{area}} \cdot f_{max}([Range(X),\ Range(Y)])}{r_0 \cdot \Theta_{scalar}} \tag{18}$$

$$\omega_{pix} = \left\lceil \frac{\theta_{dist} \cdot num(S_i)}{num(C)} \right\rceil \tag{19}$$

$$\theta_i = \begin{cases} i \cdot \omega_{pix}, & i < n \\ (i-1) \cdot \omega_{pix}, & i = n \end{cases} \tag{20}$$

where Θ_{area} is a constant.

Taking I_i as an example, if the area of a 2D boundary in I_i is smaller than θ_i, this boundary will be discarded, which means all pixels belonging to this boundary will be converted to invalid pixels. Then, we get a candidate 2D building set of I_i, $R^i = \{R_1^i, R_2^i, \ldots, R_{m_i}^i\}$, $i \in [1, n]$, with m_i denotes the number of candidate 2D buildings in R^i. Perform the above operations on all the raster images to get $R = \{R^1, R^2, \ldots, R^n\}$.

3.3. Labeling and extraction

First, the proposed constraint-based labeling method is used to detect the candidate 2D buildings belonging to the same building in the raster images, as shown in the Algorithm 1. We assume that candidate 2D buildings with the same label belong to the same building

instance. Since the point cloud reorganization might result in different numbers of candidate 2D buildings belonging to the same building in raster images. The commonly used Euclidean distance makes it difficult to distinguish whether these candidate 2D buildings belong to the same building. Therefore, we use Chebyshev distance to achieve better building instance extraction performance.

Second, duplicate pixels of candidate 2D buildings are removed according to the labels. The remaining pixels are converted to points according to the point-pixel transformation relationship retained during rasterization.

Finally, the points are grouped according to the labels to obtain the set of building instances $B = \{B_1, B_2, \ldots, B_N\}$, where $B_i \in B$ represents a building instance and N represents the number of building instances.

Algorithm 1 Constraint-based Labeling

Input:
The set of candidate 2D building sets : $R = \{R^1, ..., R^n\}$
Signle candidate 2D building : $R_j^i \in R^i$; $R_t^s \in R^s$
Distance threshold : θ_{dist}; Overlap threshold : $\theta_{overlap}$
Output: The set of labeled 2D building sets $L(R) = \{L(R^1), ..., L(R^n)\}$
Initialization:
$L_{R_j^i} = \sum_{k=1}^{i-1} m_k + j$
$L(R_j^i) = (R_j^i,\ L_{R_j^i})$
$Cent(R_j^i) = [f_{mean}(X_{R_j^i}),\ f_{mean}(Y_{R_j^i})]$
Constraints:
(a) $Distance_{chebyshev}(Cent(R_t^s), Cent(R_j^i)) \le \theta_{dist}$
(b) $Ratio_{overlap}(Rect(R_t^s), Rect(R_j^i)) \ge \theta_{overlap}$
(c) $Rect(R_t^s) \subseteq Rect(R_j^i) \vee Rect(R_t^s) \supseteq Rect(R_j^i)$
Loop:
for $1 \le i \le n-1,\ 1 \le j \le m_i$ **do**
  for $i < s \le n,\ 1 \le j \le m_s$ **do**
    if R_t^s satisfies $(a) \vee (b) \vee (c)$ **then**
      $L(R_t^s) = (R_t^s,\ L_{R_j^i})$
    end if
  end for
end for

4. Experiments

In this section, the performance of our method was evaluated and compared with several methods in both single- and dual-channel airborne LiDAR point cloud scenes. The information of test scenes is detailed in Section 4.1. Section 4.2 describes the semantic and instance evaluation metrics. Section 4.3 details the implementation information of the proposed method. In Section 4.4, we present the experimental results and analysis.

4.1. Dataset description

In this paper, six ALS point cloud scenes are used for testing, including four scenes from our own dataset and two from public datasets. DCS 1–4 were selected from the LiDAR data collected by a RIEGL VQ-1560i dual-channel system, and the LiDAR data of DCS 1–4 are raw point cloud scenes. We manually annotated all the 782,809 points of DCS 1–4 to create the ground truth. Specifically, we labeled all points to instance-level building points and non-building points (including ground, plants, cars and other non-building objects) using the open source point cloud software CloudCompare. VAIHI-1 and DALES-1 were selected from different public benchmark datasets, and their LiDAR data are building points that segmented from raw point cloud scenes (Niemeyer et al., 2014). Specifically, VAIHI-1 was from the ISPRS benchmark dataset[1] (Rottensteiner et al., 2012), which were collected with a Leica ALS50 single-channel system. DALES-1 was from the DALES datasets (Varney et al., 2020), which were obtained by a RIEGL Q1560 dual-channel system. These six test scenes differ in area, number of points, density and height. Details of these scenes are shown in Table 3.

4.2. Evaluation criteria

Since the semantic evaluation is insufficient to evaluate the performance of building extraction at instance level. Our extraction algorithm for building instances is based on both semantic and instance evaluation.

4.2.1. Semantic-based evaluation

We evaluate the semantic-level building extraction performance utilizing recall, precision and F1-score, calculated as follows:

$$Recall = \frac{TP_{point}}{TP_{point} + FN_{point}} \tag{21}$$

$$Precision = \frac{TP_{point}}{TP_{point} + FP_{point}} \tag{22}$$

$$F1\text{-}score = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision} \tag{23}$$

where TP_{point}, FP_{point} and FN_{point} are the number of true positive, false positive and false negative extracted building points, respectively.

4.2.2. Instance-based evaluation

Furthermore, we follow the instance-level evaluation of Wu et al. (2020) and Zhang et al. (2021) and same definitions of IoU (intersection over union). Since the semantic-based evaluation is not sufficient to evaluate the results of building extraction at the instance level. For each building instance, the IoU is calculated as follows:

$$IoU = \frac{point_c \cap point_g}{point_c \cup point_g} \tag{24}$$

where $point_c$ represents automatically segmented building instance point clouds and $point_g$ represents building instance point clouds in the ground truth.

In this paper, we set the minimum IoU threshold as 0.75. If IoU is greater than 0.75, the building instance is correctly extracted. When IoU is less than 0.75, if the ground truth of the building instance contains only one building instance, it is over-extracted; otherwise, it is under-extracted. We evaluate instance-level building extraction performance of point cloud scenes using IoU-based completeness, correctness, and quality (Rutzinger et al., 2009). They are calculated as follows:

$$completeness = \frac{TP_{ins}}{TP_{ins} + FN_{ins}} \tag{25}$$

$$correctness = \frac{TP_{ins}}{TP_{ins} + FP_{ins}} \tag{26}$$

$$quality = \frac{TP_{ins}}{TP_{ins} + FP_{ins} + FN_{ins}} \tag{27}$$

where TP_{ins}, FP_{ins} and FN_{ins} are the number of correctly extracted, under-extracted and over-extracted building instances in the point cloud scene, respectively.

[1] https://www.isprs.org/education/benchmarks.aspx.

Table 3
Information on the six test scenes.

Scene	Sensor type	Data type	Points	Area (m^2)	Height (m)	Density (pts/m^2)
DCS 1	Dual-channel	Raw scene	55,700	5808	15	10
DCS 2	Dual-channel	Raw scene	41,371	4813	15	9
DCS 3	Dual-channel	Raw scene	263,258	15,393	23	17
DCS 4	Dual-channel	Raw scene	422,480	23,126	30	18
VAIHI-1	Single-channel	Building points	69,562	26,753	21	–
DALES-1	Dual-channel	Building points	77,498	258,556	14	6

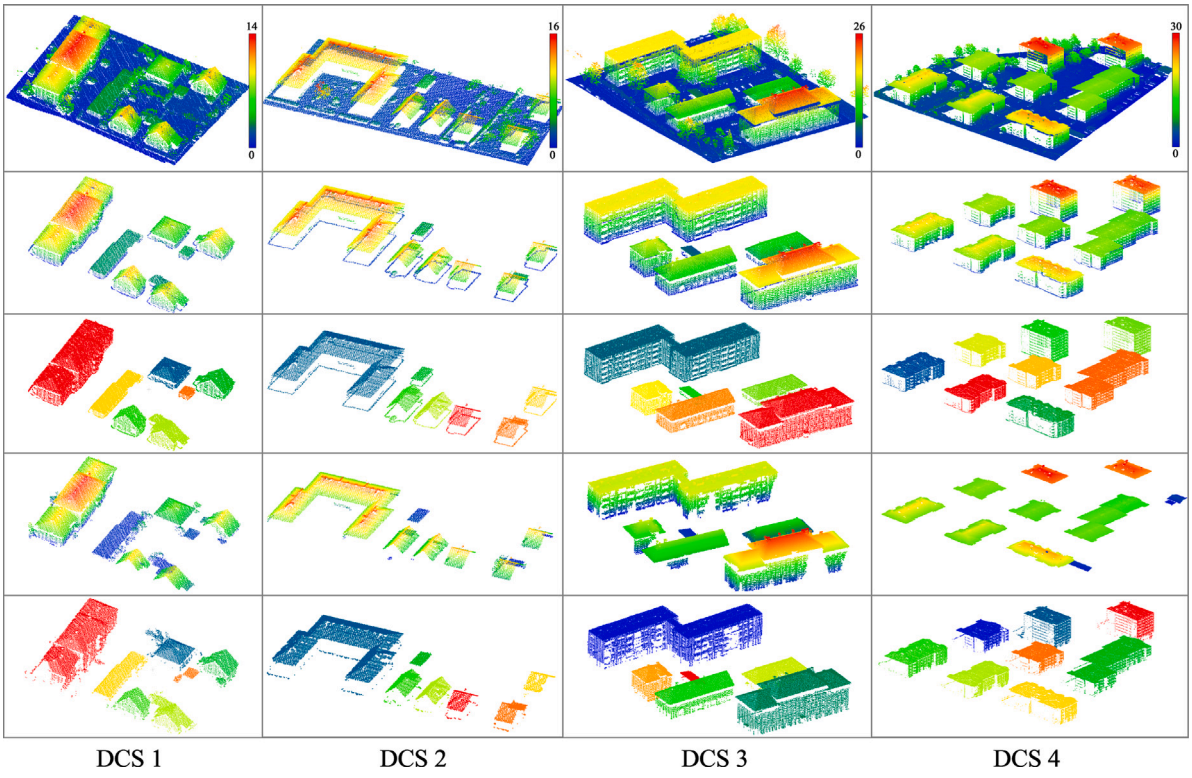


Fig. 6. Semantic-level building extraction results on DCS 1–4. 1st row: point cloud scenes. 2nd row: semantic-level ground truth. 3rd row: instance-level ground truth. 4th row: results of the LASclassify (Isenburg, 2019). 5th row: results of the proposed method. In 1st, 2nd, and 4th rows, points are displayed in different colors according to the heights. In 3rd and 5th rows, different building instances are spotted in different colors. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.3. Implementation details

The proposed method is implemented in MATLAB R2021a, Windows 10, Intel (R) Core (TM) i7-6700 CPU @ 3.40 GHz, 24.0 GB memory. In this paper, all the experiments share the same parameter settings and detailed below. During the point distribution-based slicing, we set the unit height $h_u = 0.5$ m, the stride $sd = 5$, and the repeat number $sc = 3$. Considering the heights of various objects are multiples of 0.5 m, selecting $h_u = 0.5$ m can cover the structural features of objects with different heights, which helps in more reasonably dividing the point cloud scene and retaining the structural features in the point cloud slices while avoiding redundant calculations. During the coarse 3D semantic segmentation, the point number coefficient of C is $\sigma_{num} = 10^6$, the area coefficient of C is $\sigma_{area} = \left(\frac{2}{5}\right)^2 \cdot \sigma_{num}$, and the approximate rate $\theta_{rate} = 0.7$. During the rasterization, the image basis resolution is $r_0 = 256$. During the constraint-based labeling, the overlap threshold $\theta_{overlap} = 0.4$.

4.4. Results and analysis

We evaluate the semantic-level building extraction performance with LASclassify (Isenburg, 2019) on DCS 1–4. And the instance-level performance was evaluated with several methods on VAIHI-1 and DALES-1. The instance-level performance was evaluated with several methods on VAIHI-1 and DALES-1.

Table 4
Comparison of semantic building extraction results obtained by LASclassify (Isenburg, 2019) vs. our method (%).

Scene	LASclassify			OURS		
	Recall	Precision	F1-score	Recall	Precision	F1-score
DCS 1	80.13	94.33	86.65	90.60	91.92	91.26
DCS 2	86.02	98.92	92.02	99.35	84.20	91.15
DCS 3	86.81	99.93	92.91	93.17	99.03	96.01
DCS 4	71.22	94.29	81.14	98.32	93.70	95.96
Avg.	81.04	**96.87**	88.18	**95.36**	92.21	**93.59**

4.4.1. Semantic-level building extraction results

To validate the semantic-level performance, we compare the performance of our method with LASclassify (Isenburg, 2019) on DCS 1–4. The experimental results were evaluated with recall, precision, and F1-score. The quantitative results are detailed in Table 4, showing that our method significantly outperforms LASclassify in all test scenes. Specifically, the proposed method achieves 95.36% in average recall, which reaches an improvement of 14.32% over LASclassify. In addition,

Table 5
Performance comparison of semantic-level building extraction results (%).

Scene		MV	ES2D	ES3D	LCCP	UBIS	OURS
VAIHI-1	Completeness	83.33	71.43	24.73	17.02	100.00	**100.00**
	Correctness	80.65	78.13	90.00	90.57	92.31	**92.86**
	Quality	69.44	59.52	24.06	16.72	92.31	**92.86**
DALES-1	Completeness	37.14	49.53	4.70	5.22	98.28	**100.00**
	Correctness	94.55	96.36	89.83	89.83	**100.00**	98.31
	Quality	36.36	48.62	4.67	5.19	98.28	**98.31**

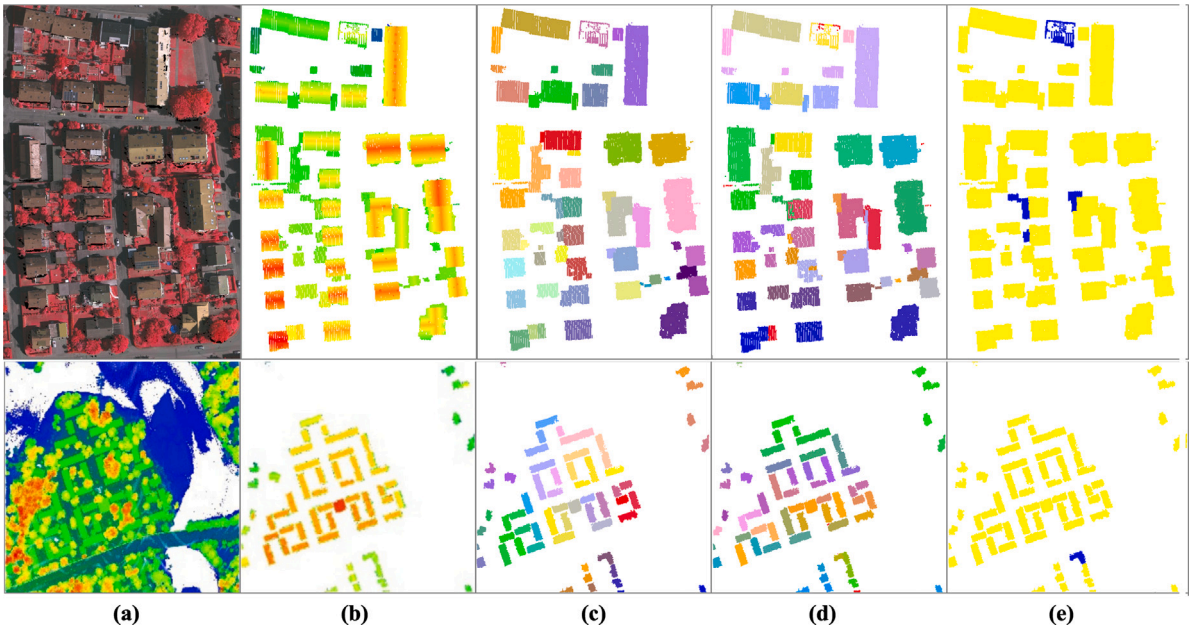


Fig. 7. Instance-level building extraction results on VAIHI-1 (first row) and DALES-1 (second row). (a) Upper: original aerial image corresponding to VAIHI-1; lower: point clouds of the entire scene corresponding to DALES-1. (b) Point clouds of VAIHI-1 (upper) and DALES-1 (lower). Points in (a) and (b) are colored by heights. (c) Instance-level ground truth. (d) Extraction results of our method. In (c) and (d), different building instances are displayed with different colors. (e) Main differences between the extraction results of our method and the ground truth (yellow: correctly extracted; blue: over-extracted). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the proposed method achieves 93.59% in average F1-score, which improves 5.41% than the 88.18% provided by LASclassify.

Benefit from the structural connection sequence and location relationships of building segments retained in the point groups. Our method has the capability to extract complete facade points of building instances. We can learn from the visualization results shown in Fig. 6 that our method achieves outstanding performance in all test scenes. The buildings extracted by our method are more complete than those extracted by LASclassify, especially for the building façades. More specifically, most of the façade points and rooftop points near the edges of the footprints were missed in the results of LASclassify, while some low and flat bushes were incorrectly extracted as buildings.

4.4.2. Instance-level building extraction results

To validate the instance-level performance, we conducted the proposed method on VAIHI-1 and DALES-1. The extraction results of our method for various types of buildings in VAIHI-1 and DALES-1 is shown in Fig. 7, with Fig. 7(e) shows the main differences between the proposed method results and the ground truth. The detailed views of the over-extracted buildings on VAIHI-1 and DALES-1 are shown in Fig. 8. It is obvious that our method achieves outstanding instance-level performance on VAIHI-1 and DALES-1.

Furthermore, we compared the proposed approach with different state-of-the-art methods for instance-level building extraction, including ES3D (Wang et al., 2015; Ramiya et al., 2017), ES2D (2D version of ES3D), MV (Sampath and Shan, 2007; Awrangjeb et al., 2013), LCCP (Stein et al., 2014), and UBIS (Zhang et al., 2021). We report the corresponding experiment results of Zhang et al. (2021) and compare them with our method in the same test scenes and evaluation metrics.

The experimental results were evaluated with instance-based evaluation. The quantified results that presented in Table 5 show that our approach outperforms the benchmark methods on VAIHI-1 and DALES-1. More specifically, for VAIHI-1, the completeness, correctness and quality of our method is better than all the others. For DALES-1, the completeness and quality of our method is higher than all the others, and the correctness higher than that of the MV, ES2D, ES3D and LCCP. Therefore, the proposed method significantly improves the instance-level performance of instance-level building extraction in different scenes. Since the proposed constraint-based labeling method can effectively distinguish candidate 2D buildings belonging to different buildings, while associating candidate 2D buildings of the same building.

5. Conclusion

This paper is the first ever to use dual-channel airborne LiDAR point clouds to extract the instance building. Dual-channel LiDAR can provide rooftops and façades simultaneously for more detailed and complete building data than traditional scanners. In addition, our novel method is the first to construct building instance extraction without preprocessing. Our proposed method consisted of (1) point cloud reorganization module, (2) rasterization and detection module, and (3) labeling and extraction module. The first module, taking advantage of a point distribution-based slicing and a top-down merging strategy,

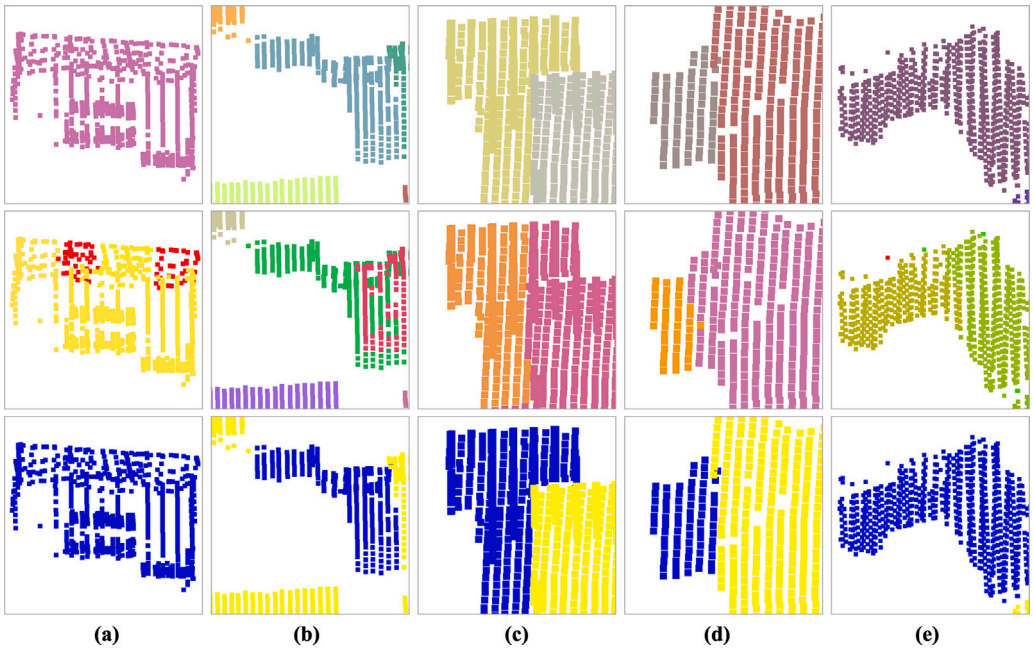


Fig. 8. Detailed views of the over-extracted buildings on VAIHI-1 and DALES-1. The corresponding instance-level ground truth, extracted results, and their differences are shown in separate rows from top to bottom. (a) to (d) belong to the over-extracted buildings on VAIHI-1 and (e) on DALES-1. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

reorganizes the raw point cloud scene into interrelated point groups. The second module rasterizes point groups into raster images for detecting 2D buildings, making use of the association between point groups and building geometric features, improving the accuracy of building detection while avoiding redundant computations. The last module utilizes a constraint-based labeling method to label 2D buildings and then extract 3D building instances. Our method has several advantages over other building extraction methods. First, the proposed method avoids preprocessing (ground filtering, noise removing or downsampling) commonly used by most existing ALS point cloud processing approaches. Therefore, the proposed method can be more automated and reliable for practical applications. Furthermore, our method can extract 3D geometric information of both rooftops and façades, which is not the case for other existing LiDAR-based building extraction methods. Experimental results on six test point cloud scenes demonstrate the outstanding performance of the proposed preprocessing-free method. For semantic-level performance, our method achieved 95.36% in average recall and 93.59% in average F1-score on four test scenes from our own dataset in the semantic-level performance. As for the instance-level performance, our approach reached 92.86% and 98.31% in quality on two test scenes from public datasets, respectively. Since the proposed method can hardly extract all buildings from point cloud scenes with large slopes, our future work will utilize the normal-based plane and slope transformation to increase the building extraction quality from such scenes.

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.

Data availability

No data was used for the research described in the article.

Acknowledgments

This work was supported by National Natural Science Foundation of China (No. 41871380).

References

Albers, B., Kada, M., Wichmann, A., 2016. Automatic extraction and regularization of building outlines from airborne lidar point clouds. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. XLI-B3, 555–560. http://dx.doi.org/10.5194/isprs-archives-XLI-B3-555-2016.

Awrangjeb, M., Fraser, C.S., 2014. Automatic segmentation of raw LiDAR data for extraction of building roofs. Remote Sens. 6 (5), 3716–3751. http://dx.doi.org/10.3390/rs6053716.

Awrangjeb, M., Zhang, C., Fraser, C.S., 2013. Automatic extraction of building roofs using LiDAR data and multispectral imagery. ISPRS J. Photogramm. Remote Sens. 83, 1–18. http://dx.doi.org/10.1016/j.isprsjprs.2013.05.006.

Cai, Z., Ma, H., Zhang, L., 2019. A building detection method based on semi-suppressed fuzzy C-means and restricted region growing using airborne LiDAR. Remote Sens. 11, 848. http://dx.doi.org/10.3390/rs11070848.

Che, E., Jung, J., Olsen, M.J., 2019. Object recognition, segmentation, and classification of mobile laser scanning point clouds: A state of the art review. Sensors 19, http://dx.doi.org/10.3390/s19040810.

Chen, D., Wang, R., Peethambaran, J., 2017. Topologically aware building rooftop reconstruction from airborne laser scanning point clouds. IEEE Trans. Geosci. Remote Sens. 55 (12), 7032–7052. http://dx.doi.org/10.1109/TGRS.2017.2738439.

Dash, J., Steinle, E., Singh, R.P., Bähr, H.P., 2004. Automatic building extraction from laser scanning data: an input tool for disaster management. Adv. Space. Res. 33 (3), 317–322. http://dx.doi.org/10.1016/S0273-1177(03)00482-4.

Deng, H., Zhang, L., Mao, X., Qu, H., 2016. Interactive urban context-aware visualization via multiple disocclusion operators. IEEE Trans. Vis. Comput. Graph. 22 (7), 1862–1874. http://dx.doi.org/10.1109/TVCG.2015.2469661.

Du, S., Zhang, Y., Zou, Z., Xu, S., He, X., Chen, S., 2017. Automatic building extraction from LiDAR data fusion of point and grid-based features. ISPRS J. Photogramm. Remote Sens. 130, 294–307. http://dx.doi.org/10.1016/j.isprsjprs.2017.06.005.

Ester, M., Kriegel, H.-P., Sander, J., Xu, X., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD. KDD '96, AAAI Press, pp. 226–231.

Fan, H., Yao, W., Tang, L., 2014. Identifying man-made objects along urban road corridors from mobile LiDAR data. IEEE Geosci. Remote Sens. Lett. 11, 950–954. http://dx.doi.org/10.1109/LGRS.2013.2283090.

Gao, J., Yang, R., 2013. Online building segmentation from ground-based LiDAR data in urban scenes. In: 3DV. IEEE, pp. 49–55. http://dx.doi.org/10.1109/3DV.2013.15.

Gonzalez, R.C., Woods, R.E., Eddins, S.L., 2004. Digital Image Processing using MATLAB. Pearson/Prentice Hall.

Huang, R., Yang, B., Liang, F., Dai, W., Li, J., Tian, M., Xu, W., 2018. A top-down strategy for buildings extraction from complex urban scenes using airborne LiDAR point clouds. Infrared Phys. Technol. 92, 203–218. http://dx.doi.org/10.1016/j.infrared.2018.05.021.

Huang, J., Zhang, X., Xin, Q., Sun, Y., Zhang, P., 2019. Automatic building extraction from high-resolution aerial images and LiDAR data using gated residual refinement network. ISPRS J. Photogramm. Remote Sens. 151, 91–105. http://dx.doi.org/10.1016/j.isprsjprs.2019.02.019.

Hui, Z., Li, Z., Cheng, P., Yevenyo, Y.Z., Fan, J., 2021. Building extraction from airborne LiDAR data based on multi-constraints graph segmentation. Remote Sens. 13, 3766. http://dx.doi.org/10.3390/rs13183766.

Isenburg, M., 2019. LAStools efficient LiDAR processing software. http://rapidlasso.com/LAStools/. Version 191018, academic.

Li, D., Shen, X., Yu, Y., Guan, H., Li, J., Zhang, G., Li, D., 2020. Building extraction from airborne multi-spectral LiDAR point clouds based on graph geometric moments convolutional neural networks. Remote Sens. 12, 3186. http://dx.doi.org/10.3390/rs12193186.

Lim, J.S., 1990. Two-Dimensional Signal and Image Processing. Englewood Cliffs.

Liu, K., Ma, H., Ma, H., Cai, Z., Zhang, L., 2020. Building extraction from airborne LiDAR data based on min-cut and improved post-processing. Remote Sens. 12, 2849. http://dx.doi.org/10.3390/rs12172849.

Maltezos, E., Doulamis, A., Doulamis, N., Ioannidis, C., 2019. Building extraction from LiDAR data applying deep convolutional neural networks. IEEE Geosci. Remote Sens. Lett. 16 (1), 155–159. http://dx.doi.org/10.1109/LGRS.2018.2867736.

Meng, X., Wang, L., Currit, N., 2009. Morphology-based building detection from airborne lidar data. Photogramm. Eng. Rem. Sens. 75, 437–442. http://dx.doi.org/10.14358/PERS.75.4.437.

Nguyen, T.H., Daniel, S., Guériot, D., Sintes, C., Caillec, J.-M.L., 2020. Super-resolution-based snake model - an unsupervised method for large-scale building extraction using airborne LiDAR data and optical image. Remote Sens. 12, 1702. http://dx.doi.org/10.3390/rs12111702.

Niemeyer, J., Rottensteiner, F., Soergel, U., 2014. Contextual classification of lidar data and building object detection in urban areas. ISPRS J. Photogramm. Remote Sens. 87, 152–165. http://dx.doi.org/10.1016/j.isprsjprs.2013.11.001.

Pu, S., Vosselman, G., 2009. Knowledge based reconstruction of building models from terrestrial laser scanning data. ISPRS J. Photogramm. Remote Sens. 64, 575–584. http://dx.doi.org/10.1016/j.isprsjprs.2009.04.001.

Qin, R., Huang, X., Gruen, A., Schmitt, G., 2015. Object-based 3-D building change detection on multitemporal stereo images. IEEE J. Sel. Top Appl. Earth Obs. Remote Sens. 8 (5), 2125–2137. http://dx.doi.org/10.1109/JSTARS.2015.2424275.

Ramiya, A.M., Nidamanuri, R.R., Krishnan, R., 2017. Segmentation based building detection approach from LiDAR point cloud. Egypt. J. Remote Sens. Space Sci. 20 (1), 71–77. http://dx.doi.org/10.1016/j.ejrs.2016.04.001.

RIEGL, 2019. RIEGL VQ-1560i data sheet. http://www.riegl.com/.

Rottensteiner, F., Sohn, G., Jung, J., Gerke, M., Baillard, C., Benitez, S., Breitkopf, U., 2012. The ISPRS benchmark on urban object classification and 3D building reconstruction. ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci. 1 (1), 293–298. http://dx.doi.org/10.5194/isprsannals-I-3-293-2012.

Rutzinger, M., Höfle, B., Oude Elberink, S., Vosselman, G., 2011. Feasibility of facade footprint extraction from mobile laser scanning data. Photogramm. Fernerkun. 2011, 97–107. http://dx.doi.org/10.1127/1432-8364/2011/0075.

Rutzinger, M., Rottensteiner, F., Pfeifer, N., 2009. A comparison of evaluation techniques for building extraction from airborne laser scanning. IEEE J. Sel. Top Appl. Earth Obs. Remote Sens. 2, 11–20. http://dx.doi.org/10.1109/JSTARS.2009.2012488.

Sampath, A., Shan, J., 2007. Building boundary tracing and regularization from airborne LiDAR point clouds. Photogramm. Eng. Remote Sens. 73 (7), 805–812. http://dx.doi.org/10.14358/PERS.73.7.805.

Stein, S.C., Wörgötter, F., Schoeler, M., Papon, J., Kulvicius, T., 2014. Convexity based object partitioning for robot applications. In: ICRA. IEEE, pp. 3213–3220. http://dx.doi.org/10.1109/ICRA.2014.6907321.

Varney, N., Asari, V.K., Graehling, Q., 2020. Dales: A large-scale aerial lidar data set for semantic segmentation. In: CVPRW. IEEE/CVF, pp. 186–187. http://dx.doi.org/10.1109/CVPRW50498.2020.00101.

Wang, Y., Jiang, T., Yu, M., Tao, S., Sun, J., Liu, S., 2020. Semantic-based building extraction from LiDAR point clouds using contexts and optimization in complex environment. Sensors 20, http://dx.doi.org/10.3390/s20123386.

Wang, H., Zhang, W., Chen, Y., Chen, M., Yan, K., 2015. Semantic decomposition and reconstruction of compound buildings with symmetric roofs from LiDAR data and aerial imagery. Remote Sens. 7 (10), 13945–13974. http://dx.doi.org/10.3390/rs71013945.

Wu, C.-Y., Hu, X., Happold, M., Xu, Q., Neumann, U., 2020. Geometry-aware instance segmentation with disparity maps. In: CVPRW. IEEE/CVF, http://dx.doi.org/10.48550/arXiv.2006.07802.

Xia, S., Wang, R., 2018. Extraction of residential building instances in suburban areas from mobile LiDAR data. ISPRS J. Photogramm. Remote Sens. 144, 453–468. http://dx.doi.org/10.1016/j.isprsjprs.2018.08.009.

Xie, Y., Tian, J., Zhu, X.X., 2020. Linking points with labels in 3D: A review of point cloud semantic segmentation. IEEE Geosci. Remote Sens. Mag. 8 (4), 38–59. http://dx.doi.org/10.1109/MGRS.2019.2937630.

Yang, B., Wei, Z., Li, Q., Li, J., 2013a. Semiautomated building facade footprint extraction from mobile LiDAR point clouds. IEEE Geosci. Remote Sens. Lett. 10 (4), 766–770. http://dx.doi.org/10.1109/LGRS.2012.2222342.

Yang, B., Xu, W., Dong, Z., 2013b. Automated extraction of building outlines from airborne laser scanning point clouds. IEEE Geosci. Remote Sens. Lett. 10, 1399–1403. http://dx.doi.org/10.1109/LGRS.2013.2258887.

Yu, B., Liu, H., Wu, J., Hu, Y., Zhang, L., 2010. Automated derivation of urban building density information using airborne LiDAR data and object-based method. Landsc. Urban Plan. 98 (3), 210–219. http://dx.doi.org/10.1016/j.landurbplan.2010.08.004.

Yuan, Q., Shafri, H.Z.M., Alias, A.H., bin Hashim, S.J., 2021. Multiscale semantic feature optimization and fusion network for building extraction using high-resolution aerial images and LiDAR data. Remote Sens. 13, 2473. http://dx.doi.org/10.3390/rs13132473.

Zhang, P., Du, P., Lin, C., Wang, X., Li, E., Xue, Z., Bai, X., 2020. A hybrid attention-aware fusion network (HAFNet) for building extraction from high-resolution imagery and LiDAR data. Remote Sens. 12, 3764. http://dx.doi.org/10.3390/rs12223764.

Zhang, Y., Yang, W., Liu, X., Wan, Y., Zhu, X., Tan, Y., 2021. Unsupervised building instance segmentation of airborne LiDAR point clouds for parallel reconstruction analysis. Remote Sens. 13, 1136. http://dx.doi.org/10.3390/rs13061136.

Zhou, Z., Gong, J., 2018. Automated residential building detection from airborne LiDAR data with deep neural networks. Adv. Eng. Inf. 36, 229–241. http://dx.doi.org/10.1016/j.aei.2018.04.002.