Taylor & Francis
Taylor & Francis Group

International Journal of Remote Sensing

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/tres20

# Object-based building instance segmentation from airborne LiDAR point clouds

Wangshan Yang, Xinyi Liu, Yongjun Zhang, Yi Wan & Zheng Ji

**To cite this article:** Wangshan Yang, Xinyi Liu, Yongjun Zhang, Yi Wan & Zheng Ji (2022) Object-based building instance segmentation from airborne LiDAR point clouds, International Journal of Remote Sensing, 43:18, 6783-6808, DOI: 10.1080/01431161.2022.2145582

**To link to this article:** https://doi.org/10.1080/01431161.2022.2145582

Published online: 28 Nov 2022.

INTERNATIONAL JOURNAL OF REMOTE SENSING
2022, VOL. 43, NO. 18, 6783–6808
https://doi.org/10.1080/01431161.2022.2145582

# Object-based building instance segmentation from airborne LiDAR point clouds

Wangshan Yang, Xinyi Liu, Yongjun Zhang, Yi Wan and Zheng Ji

School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, China

**ABSTRACT**
Building instance segmentation is of very importance to parallel reconstruction, management and analysis of building instance. Previous studies of building instance segmentation mainly focused on the building scenes where the building spacing is much larger than the point spacing, while the accuracy of building instance segmentation for complex buildings scenes and the building point clouds where the space between buildings is similar with point spacing is low. To improve the accuracy of building instance segmentation for complex building scenes, we propose a novel object-based building instance segmentation (OBBIS) method from airborne light detection and ranging (LiDAR) point clouds. Firstly, our proposed method divides building point clouds into objects, and then the objects are classified according to the characteristics of building roof plane objects, roof accessory objects and building facade objects. Secondly, we use node to represent object and then a fix-size feature vector is inferred for each node. Thirdly, vertical cylinder neighbour node graph is constructed. Finally, the energy function is constructed according to the relationship between the nodes, and then the objects are merged according to the energy minimum (that is, objects are merged with a minimum energy to obtain the building instances). Comprehensive experiments on benchmark datasets demonstrate that the proposed OBBIS method performs better than eight state-of-the-art building instance segmentation methods.

**ARTICLE HISTORY**
Received 7 July 2022
Accepted 3 November 2022

**KEYWORDS**
airborne LiDAR point clouds; building instance segmentation; object; model consistency evaluation

## 1. Introduction

Instance segmentation produces different labels for different object instances that belong to the same class (Hafiz and Bhat 2020). In recent years, with the rapid development of photogrammetric technique, it has become possible to quickly obtain a three-dimensional model of the earth’s surface in a short period of time. However, the current 3D model is a skin of the terrain surface, and each object cannot be queried, processed and analysed separately, which limits the application of the 3D model. According to the building instance segmentation methods, building point clouds are segmented into building instances. Each building instance can be individually selected and added attributes, which provides the possibility for building management, query, analysis, and parallel reconstruction. Building instance segmentation plays an important role in point cloud based building modelling. At present, most of the

**CONTACT** Xinyi Liu liuxy0319@whu.edu.cn; Yongjun Zhang zhangyj@whu.edu.cn School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, China

research on building instance segmentation mainly focused on the building scenes where the building spacing is much larger than the point cloud point spacing (Yang et al. 2013), as shown in Figure 1(a) and a small amount of building spacing is close to the point cloud point spacing (Du et al. 2019), as shown in Figure 1(b). For complex building scenes and the building point clouds where the space between buildings is similar with point spacing, it is difficult to individually separate adjacent buildings, as shown in Figure 1(a). Query statistics and parallel reconstruction of buildings (each building uses one thread to model, and the GPU can call multiple threads at one time to perform 3D reconstruction of the buildings in the scene at the same time) will get affected if these buildings are not segmented into building instance. To meet the needs of the subject query, reconstructed three-dimensional building models must be physically distinguishable. Building instance is defined as follows: the protruding ground

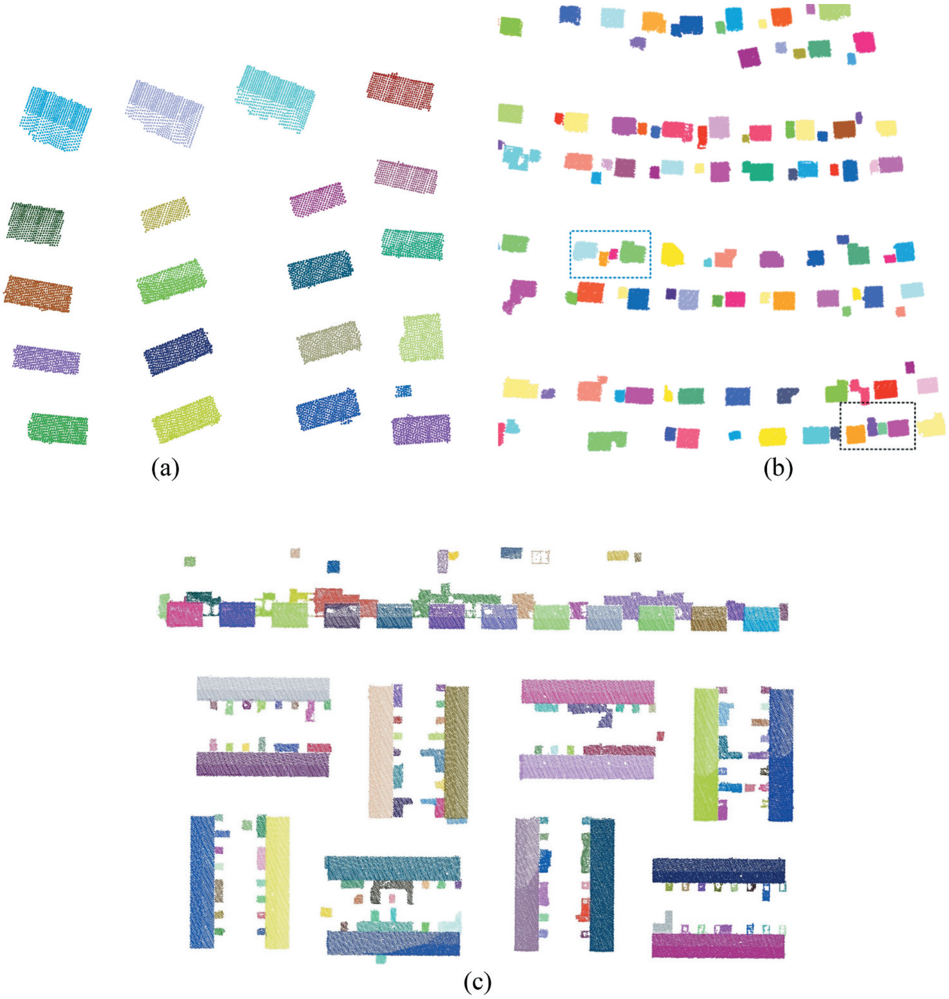


Figure 1. Building point clouds: (a) the building spacing is much larger than the point cloud point spacing; (b) a small amount of building spacing is close to the point cloud point spacing; (d) complex buildings scenes and the building point clouds where the space between buildings is similar with point spacing.

part does not have a common set (common connection part) with other buildings and cannot be distinguished on computer vision (Zhang et al. 2021).

Light Detection and RangingLiDAR() is an active remote sensing system that is less affected by weather conditions such as light and can quickly and real-time collect three-dimensional surface information of the ground or ground objects. The data is an important source for obtaining the information on the earth's surface (Zeng, Mao, and Li et al. 2007), which is mainly used in point cloud filtering (Zhang and Lin 2013a; Zhao et al. 2016b; Kang, Liu, and Lin 2014; Sithole and Vosselman 2004), point cloud classification (Hackel, Savinov, and Ladicky et al. 2017; A, H, and S et al. 2019; Liu et al. 2020), road boundary extraction (Yang, Cai, and Gu 2018; Liu, Ma, and Lu 2022), building extraction (Zhao, Duan, and Zhang et al. 2016a), Building modelling (Liu et al., 2019b; Xiao, Wang, and Li et al. 2015; Cao et al. 2017; Filin and Pfeifer 2005; Haala nd Kada 2010; Wang, Lindenbergh, and Menenti 2018; Moisan, Heinkele, and Foucher et al. 2021) and tree instance segmentation (Liang et al. 2018; Jing, Hu, and Li et al. 2012; Lee, C, and E et al. 2010; Grigorijs, Stefan, and Shaun et al. 2018; Claudia and Davide et al. 2016; Lee, Cai, and Lellmann et al. 2017). The research on building instance segmentation has also attracted extensive attention in recent years (Huang, Cao, and Cao 2018; Ural and Shan 2020).

At present, although the existing instance segmentation methods provide satisfactory building instance segmentation results, they still have some limitations. Most of them (e.g. the Euclidean segmentation method, the voxel segmentation method, the moving window algorithm and the region growing method) perform well in building point cloud scenes where the building spacing is much larger than the point cloud point spacing. However, existing building instance segmentation methods often produce under-segmentation for complex buildings scenes. Complex building instances are connected by building facades or walls. The building roof is generally composed of one plane or multiple planes, and there may be complex topologies and convex-concave relationships. It is difficult to determine which planes belong to the same building instance based on the topological relationship and convexity between the planes if we first extract the roof plane and then judge topological relationship and convex-concave relationship of the roof plane. For complex building scene, how to have a better building point cloud instance segmentation is still a huge challenge. Since the points are sparse, scattered and unevenly distributed, point-based point cloud segmentation is easily disturbed by noise and outliers, resulting in segmentation errors (Lee and Schenk 2002; Filin and Pfeifer 2005; Jonathan et al. 2001; Zhang, Lin, and Liang 2017; Dong et al. 2018). Voxel-based point cloud segmentation is to divide the point clouds into cubes of a certain voxel size (Wang, Lindenbergh, and Menenti 2018; Wang and Tseng 2011). All points in each cube are a voxel, and voxel features are extracted according to all points within the voxel. However, voxel segmentation also has some disadvantages. Point clouds in the same voxel will contain different instances point clouds or different semantic point clouds if the size of the voxel is too large. The local feature value will be greatly affected by noise if the setting is too small (Papon et al. 2013; Yang, Cai, and Gu 2018; Dong et al. 2018; Liu et al. 2020). Object-based point cloud segmentation is to segment the point clouds into point cloud clusters, and segmented point cloud clusters belonging to the same object are assigned the same number. Each object is treated as a processing unit, and each object is geometrically simple and semantically homogeneous cluster (that is, all the points of the cluster that are geometrically simple and semantically homogeneous belong to the same instance, and each cluster does not contain objects from different instances) (Zhang and Lin 2013a; Zhang, Lin, and Ning 2013b; Lin and Zhang 2014). This paper proposes a new object-based building instance segmentation (OBBIS) method from airborne LiDAR point clouds. The main contributions of the OBBIS method are as follows:

(1) We propose an object classification method to classify objects into facade objects, roof plane objects and roof accessory objects, which can support merging of objects.
(2) We propose a cylinder neighbour nodes relationship graph construction method, which enables each node to establish a relationship with other nodes in the neighbour of the cylinder of the node.
(3) We propose an energy minimization-based object merging method to achieve building instance segmentation by cylinder neighbour node relationship graph between objects.

## 2. Related work

There are many studies in instance segmentation for various data sources (Xu et al. 2021). In this section, we review the existing works of building instance segmentation for 2D remote sensing images and 3D point cloud.

Building instance segmentation for 2D remote sensing images. In recent years, deep learning methods had been extensively studied from a variety of aspects, such as image segmentation, point cloud classification and point cloud filtering. Existing building instance segmentation methods for remote sensing images mainly focused on deep learning methods. He et al. (2017) proposed Mask R-CNN instance segmentation algorithm. Zhao et al. (2016b) used the Mask R-CNN algorithm to achieve building instance segmentation for satellite images. Iglovikov et al. (2018) proposed a TernausNetV2 - a simple fully convolutional network to realize building instance segmentation from high-resolution satellite imagery. Ji et al. (2019a) proposed a Siamese fully convolutional network model that obtained better segmentation accuracy. Ji, Wei, and Lu (2019b) proposed a novel convolutional neural network (CNN)-based change detection framework for locating changed building instances as well as changed building pixels from very high resolution (VHR) aerial images. Li et al. (2020) proposed a building instance segmentation network for in high-resolution remote sensing images. The method firstly detects key points of a building and then reconstructs semantic masks with these key points so that the sharp boundary of the building could be preserved. Deshapriya et al. (2020) demonstrated the feasibility of the proposed architecture with respect to instance segmentation tasks on satellite images, which have a wide range of applications. The building instance segmentation of remote sensing images is of great significance to the study of building point cloud instance segmentation. The segmented remote sensing image building instances are projected into 3D space and then are regarded as masks to achieve building point clouds instance segmentation.

Building instance segmentation for 3D point clouds. Existing building point clouds instance segmentation methods mainly include Euclidean clustering algorithm, moving window algorithm, traditional connected-component labelling algorithm, density-based spatial clustering of applications with noise (DBSCAN) algorithm and some density-based optimization algorithms (Wang et al. 2020a, 2020b; Ural and Shan 2020). In recent years, many researchers have used the Euclidean clustering method to segment the building point clouds into individual clusters (Ramiya, Nidamanuri, and Krishnan 2017; Wang et al. 2015; Gamal et al. 2020). The method can better achieve building instance segmentation for building scenes with building spacing much larger than point spacing, but for connected buildings, there are many under segmentation. To address this problem, a moving window algorithm was used to divide filtered LiDAR point clouds into point cloud clusters, and each point cloud cluster represents an individual building or a tree. Then, the set of trees was removed from the point cloud cluster (Mohammad and Clive 2014). Yan and Wei (2018) proposed a building instance segmentation method for dense matching point clouds. Specifically, roof point clouds are extracted and then the roof point clouds are projected into a two-dimensional (2D) grid. Each building instance point clouds was obtained according to the topological relationship between the grids and thereby realizes the building instance segmentation. These methods can segment partially connected buildings, while fully connected buildings are difficult to segment into separate building instances. The traditional connected-component labelling (TCCL) algorithm (Sohn and Dowman, 2007) was embedded in the open-source software package CloudCompare (Du et al. 2019). Lehtomaki et al. (2016) proposed a voxel of a 3D grid to segment building point clouds to building instance. The method first voxelizes the point clouds and initialize a segmentation, and then selects one voxel as seed voxel. Then, the seed's occupied 26-neighbours are retrieved and added to the segment. This process continues until all voxels are treated and belong to some segment. Axel and Aardt (2017) proposed a region-growing-with-smoothness-constraint method for building instance segmentation. Cao et al. (2017) and Huang, Cao, and Cao (2018) used density-based spatial clustering of applications with noise (DBSCAN) to segment building point clouds into building instance (Kurdi and Awrangjeb 2020). Du et al. (2019) used proposed an enhanced DBSCAN algorithm to segment building point clouds to building instance. Wang et al. (2020a) proposed a convolution-based filter DBSCAN building instance method. The method averages the heights of the k nearest neighbours of point *Pi* through convolution filtering, increases the distance between buildings, and then uses the DBSCAN algorithm to segment the building point clouds into building instances. Wang et al. (2020b) proposed a building instance segmentation method that combines the Random Sample Consensus (RANSAC) algorithm and the DBSCAN algorithm. The method first extracts planes from building point clouds by RANSAC algorithm, and then applies the DBSCAN algorithm to separate individual buildings from dense buildings. Ural and Shan (2020) proposed DBSCAN-clustering-with-normal-vector -constraint approach. The method first calculate the surface normal for each points to obtain the angular divergence of each point's surface normal from the horizontal vector at that point. Each point with a divergence angle smaller than a threshold of 10º is then removed from the dataset. Next, we use DBSCAN clustering algorithm to label building instance.

## 3. Method

The proposed OBBIS method achieves building instance segmentation by segmenting building point clouds into objects and then merging objects of the same building instance together. The input data consists of a set of points that has been classified as buildings. Our OBBIS method includes object extraction, object features extraction,

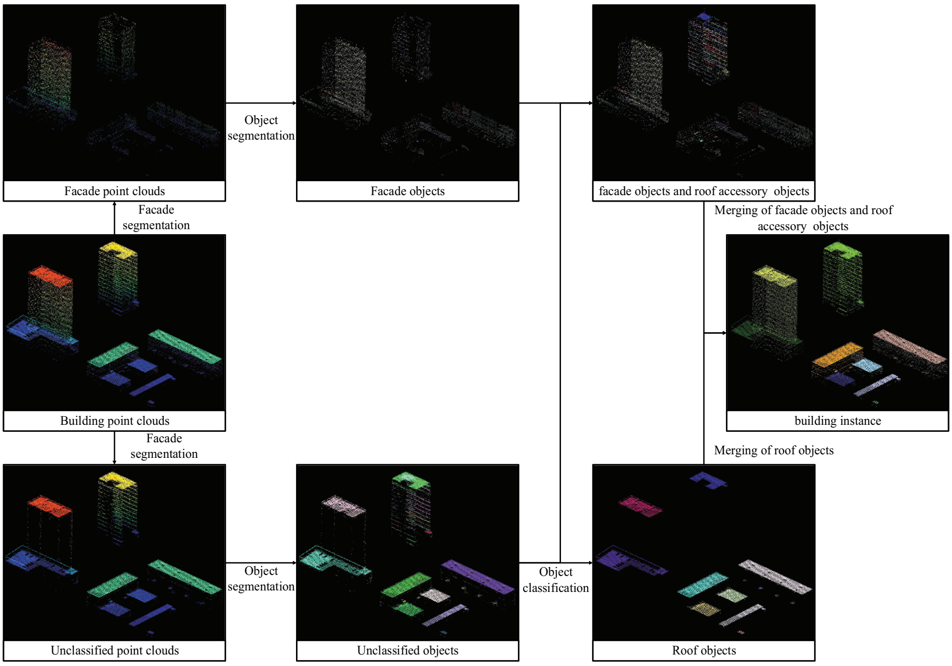


**Figure 2.** Illustration of the proposed OBBIS method.

vertical cylinder neighbour node graph and object merging. Figure 2 illustrates the pipeline of the proposed OBBIS method.

## 3.1. Object extraction

Object extraction includes facade segmentation, object segmentation and object classification. Firstly, according to the linear, planar and spherical features of the closest neighbour points of each point the normal vector feature of the plane, the building point clouds are divided into facade point clouds and unclassified point clouds (that is, point clouds in the building point clouds that are not labelled as facade point clouds, roof plane point clouds or roof accessory point clouds.). And then, classified facade point clouds and building unclassified point clouds are segmented into building facade objects and building unclassified objects, respectively. Finally, building unclassified objects are further classified into facade objects, roof plane objects and roof accessory objects.

### 3.1.1. Facade segmentation

In order to find the closest neighbour points for a given 3D point, the most commonly used approach is based on a KD-tree (Friedman, Bentley, and Finkel 1977; Weinmann et al. 2015; Liu et al. 2019). Generally, a KD-tree represents a compact hierarchical data structure for point sets and thus the nearest neighbour points can be quickly searched. KD -tree is created for building point clouds by the point cloud library (PCL). Principal component analysis is used to fit the plane of the k-nearest point set of each point, and then the three eigenvalues ($\lambda_1$, $\lambda_2$ and $\lambda_3$) of their covariance matrix are their fittings squared differences in the three directions (Nahr and Saadatseresht 2021). According to the covariance, linear, planar, and volumetric features, building point clouds are identified ($v_l = 1$ k-nearest point set of the point is volumetric, as shown in Figure 3, and the range of elevation is 0–56.045 m; $v_l = 2$ k-nearest point set of the point is planar, as shown in Figure 4, and the range of elevation is 0–56.323 m; $v_l = 3$k-nearest point set of the point is linear, as shown in Figure 5, and the range of elevation is 0–56.27 m.). Figure 6 shows volumetric points, planar points and linear points, which are red, yellow and blue, respectively. The linear $\alpha_{1d}$, planar $\alpha_{2d}$ and volumetric $\alpha_{3d}$ dimensional category $v_l$are calculated as Equation (1) and Equation (2), respectively.

$$\alpha_{1d} = \frac{\sqrt{\lambda_1} - \sqrt{\lambda_2}}{\sqrt{\lambda_1}}, \alpha_{2d} = \frac{\sqrt{\lambda_2} - \sqrt{\lambda_3}}{\sqrt{\lambda_1}}, \alpha_{3d} = \frac{\sqrt{\lambda_3}}{\sqrt{\lambda_1}} \tag{1}$$

$$V_l = arg_{x\in[1,3]}max(\alpha_{xd}) \tag{2}$$

where $\lambda_1, \lambda_2, \lambda_3$ ($\lambda_1 \geq \lambda_2 \geq \lambda_3$) are eigenvalue of the covariance matrix by the k-nearest point set of each point. For planar point clouds, normal vector $V_n$ feature (the feature vector corresponding to the smallest eigenvalue $\lambda_3$ of the covariance matrix) of k-nearest point set of the point is extracted. At the same time, angle $\theta_1$ between plane normal vector and vertical direction is calculated, which is used to judge whether the point belongs to facade as Equation (3) and Equation (4).

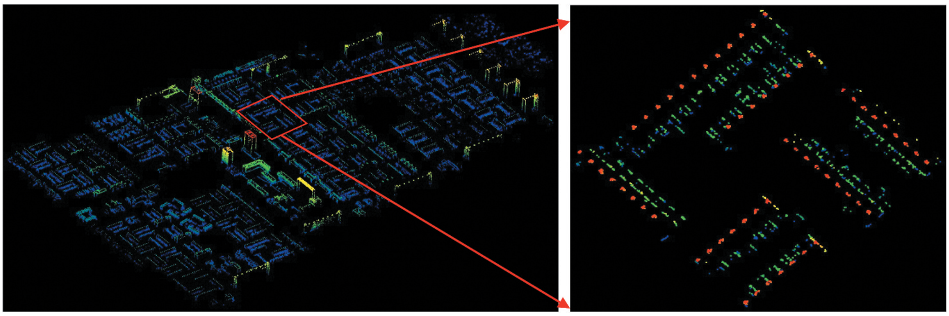


Figure 3. Volumetric points.

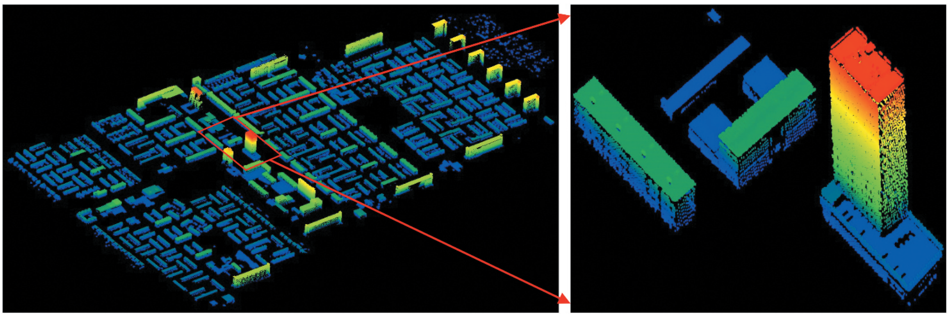


Figure 4. Planar points.

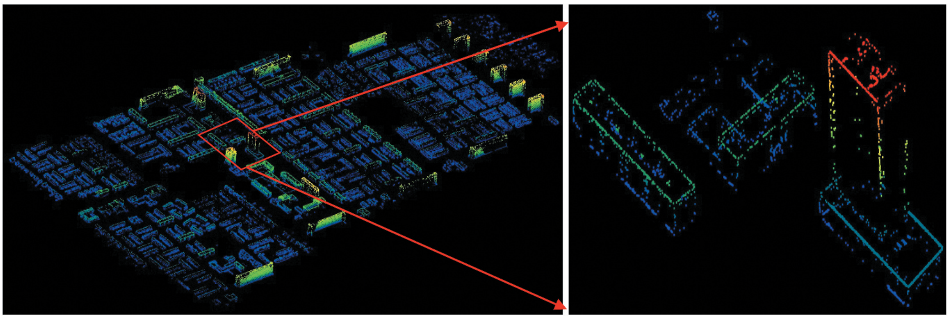


Figure 5. Linear points.

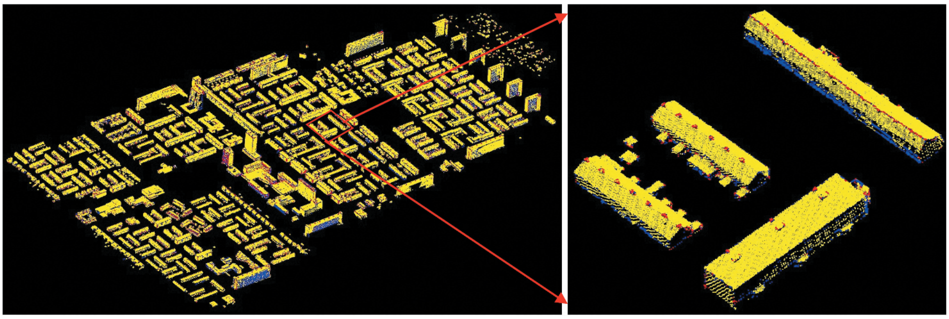


Figure 6. Volumetric points, planar points and linear points.

$$\theta_1 - 90^\circ \le \theta_2 (90^\circ \le \theta_1 \le 180^\circ) \tag{3}$$

$$90^\circ - \theta_1 \le \theta_2 (0^\circ \le \theta_1 \le 90^\circ) \tag{4}$$

where $\theta_2$ is an angle threshold between building facade point clouds normal vector and horizontal plane normal vector. The point belongs to facade point if Equation (3) or Equation (4) is true. Otherwise, it is building unclassified point clouds. K-nearest point sets of the point is linear or volumetric, which is identified as building unclassified point clouds.

### 3.1.2. Object segmentation

The object segmentation method is to segment point clouds into geometrically simple and semantically homogeneous clusters (i.e. all the points of the cluster in a geometrically simple and semantically homogeneous belong to the same instance and each part does not contain objects from different instances). In this section, we use object segmentation method to segment building facade point clouds and building unclassified point clouds into building facade objects and building unclassified objects, respectively. According to the density information and neighbour information of point cloud, we propose an object segmentation method that can segment connected multiple buildings into different objects. The pseudocode of the proposed object segmentation algorithm is shown below in Algorithm1.

Algorithm1 (proposed object segmentation algorithm)

```
Notation:
$bc$ : Building point clouds;
$dc_1$: Neighbour radius of selected point;
$num_1$: The minimum the number of points within the neighbour radius of every point;
$num$: The number of points in the neighbour radius of the selected point;
Input: $bc$,$dc_1$,$num_1$;
Output: object;
All points are marked as false
For i=1, 2, . . . , m, do
  Select the $i_{th}$ point p marked as false and then mark as true
  Calculate points number $num$ in the neighbour radius $dc_1$ of the point
  If $num < num_1$
    The point within the neighbour radius $dc_1$ are regard as an object
  Else
    The point within the neighbour radius is added into the object, and then the points marked as false are regarded as object centre
  For j=1, 2, . . . , m, do
    Take the point that is not marked as true in the object centre as centre, and add the point within neighbour radius into object centre
    End for
  End if
End for
Output: object
```

### 3.1.3. Object classification

The point cloud normal vector is calculated by fitting the plane of k neighbour point sets, and then the building facade point clouds and unclassified point clouds are segmented according to plane normal vector. Due to the limitations of this facade segmentation method, unclassified point clouds are composed of facade point clouds, roof plane point clouds and roof accessory point clouds. In this paper, we propose an object classification method to classify the unclassified objects into facade objects, roof plane objects and roof accessory objects.

For unclassified objects, each unclassified object is marked as unvisited, and unclassified object is converted to two-dimensional point clouds (i.e. the z value of point clouds is 0). The PCL is used to create a two-dimensional KD-tree for the two-dimensional point clouds. Each point is processed and then the number of the unclassified object point clouds and the number of the object point clouds are counted in the cylinder neighbour radius of each point. In each object, the number $num_u$ of the unclassified object point clouds and the number $num_o$ of the object point clouds are counted. The ratio $num_r$ is the number of unclassified object point clouds $num_u$ to the number of object point clouds $num_o$ as Equation (5).

$$num_r = \frac{num_u}{num_o} \tag{5}$$

where the object is a building facade object if the ratio $num_r$ is greater than the threshold. Otherwise, labels of the unclassified objects and labels of the objects in the cylinder neighbour radius of the unclassified objects are counted. The maximum value ($max_{x1}$, $max_{y1}$, $max_{z1}$), minimum value ($min_{x1}$, $min_{y1}$, $min_{z1}$) of each unclassified object and the maximum value ($max_{x2}$, $max_{y2}$, $max_{z2}$), minimum value ($min_{x2}$, $min_{y2}$, $max_{z2}$) of all the neighbour object of each object are calculated. The object is roof accessory objects if

Equation (6), Equation (7) and Equation (8) are all true at the same time. Otherwise, it is roof plane object.

$$((max_{x2} + buffer_m) > max_{x1})\&\&((min_{x2} - buffer_m) < min_{x1})) \tag{6}$$

$$((max_{y2} + buffer_m) > max_{y1})\&\&((min_{y2} - buffer_m) < min_{y1})) \tag{7}$$

$$((max_{z2} - min_{z1}) > height_{min})\&\&((max_{z1} - max_{z2}) < height_{max}) \tag{8}$$

where $buffer_m$ is buffer threshold, it is equal to the point spacing of point clouds. The maximum height difference threshold and the minimum difference threshold between the maximum of the roof accessory objects and the maximum height of the building roof objects are $height_{max}$ and $height_{min}$.

### 3.2. Object features extraction

Each object represents a building instance or a part of a building instance. In this section, we extract a descriptor for each object, and the descriptor is a 4D feature vector (i.e. model consistency value, object model consistency difference value, two-dimensional distance between centre object and other objects within cylinder neighbour radius of the centre object, and object classification information), where object classification information refers to whether each object is a roof plane object, a facade object or a roof accessory object. By doing so, each object has certain semantic information, which will later support object merging.

Based on the visual psychology of Gestalt theory, the factors that can attract human visual attention to some basic structures include colour constancy law, vicinity law, similarity law, Rubin's closure law, constant width law, symmetry law and convexity law (Delsolneux, Moisan, and Morel 2008). The height of the same building roof is similar and its density is uniform. Based on the characteristics of Gestalt theory and buildings, a building is regarded as a column structure. Due to the building facade under the building roof, the merged building roof and facade is consistent with the column structure model. The model consistency method (Zhang et al. 2021) is used to calculate the model consistency value of each object.

In order to obtain model consistency difference value of object, the object is regarded as the centre object $csp_1$, and object $csp_2$ within cylinder neighbour radius of the centre object $csp_1$ is with more points than the central object $csp_1$. Model consistency value $rt_2$ of the object $csp_2$ is calculated. And then, the object $csp_1$ is merged into the object $csp_2$, and the model consistency value $rt_2^{'}$ is calculated. The model consistency difference value $df$ between $rt_2$ and $rt_2^{'}$ is calculated as Equation (9).

$$df = rt_2 - rt_2^{'} \tag{9}$$

For the extraction of the two-dimensional distance between centre object and other objects within cylinder neighbour radius of the centre object, a certain object point clouds within cylinder neighbour radius of the centre object are converted to a two-dimensional point clouds (z equals to 0). And then, PCL is used to create KD-tree. Each point of the central object is regarded as centre point (z equals to 0) to find the nearest point in the KD-tree and then record the minimum two-dimensional distance $dis_{pq}$ among them.

Finally, the minimum value $dis_{pqmin}$ of the minimum two-dimensional distance $dis_{pqmin}$ in the object is calculated.

### 3.3. Cylinder neighbour node relationship graph construction

Each object is treated as a node before cylinder neighbour node relationship graph is constructed, and then object features are embedded into node. Each node has certain feature information (facade objects, roof plane objects, roof accessory objects, model consistency value, model consistency difference value, two-dimension distance between objects), but there is no topological information between neighbour nodes. The current node relationship graph construction methods include K-Nearest Neighbour (KNN) (Ben-Shabat et al. 2017). The KNN node graph construction method links each node to the nearest k nodes. Since building facade point clouds are relatively sparse and building facade point clouds are divided into many objects so the building also has many nodes. However, some buildings have fewer facade objects, the building has fewer nodes. Therefore, it is difficult to set the threshold k of KNN node graph construction. Delaunay triangulation conforms to a similar principle.

To solve the above problem, we propose a novel cylinder neighbour node graph construction method. Each node constructs a neighbour node graph within its cylinder neighbour radius with other nodes. The cylinder neighbour node graph construction process is a structured representation of the point cloud, and relationship between nodes is constructed by cylinder neighbour node graph. A cylinder neighbour node graph$G = (S, E, F)$ consists of objects $S$ (i.e. nodes), edges $E$ (i.e. adjacency between nodes), and node feature $F$ (i.e. model consistency value, object model consistency difference value, two-dimensional distance between centre node and other nodes within cylinder neighbour radius of the centre node, and object classification information). The pseudocode of cylinder neighbour node graph construction is shown below in Algorithm 2.

Algorithm 2 (cylinder neighbour node graph construction algorithm)

```
Notation:
s: Object;
E: edge;
$dc_2$: Cylinder neighbour radius of selected object;
$num_2$: The number of objects in the cylinder neighbour radius of centre object;
Input: s, $dc_2$, $num_2$;
Output: cylinder neighbour node graph;
For i = 1, 2, ..., m, do
  Select the ith object $s$, which is regarded as a node
  Each point of the node is used as the search centre, $num_2$is calculated
  If $num_2 > 1$
    cylinder neighbour node graph is constructed between centre node and nodes within the cylinder neighbour radius of the centre node, nodes are linked by edges $E$. Each node contains feature vector of the node
  Else
    the node is an isolated node
  End if
End for
Output: cylinder neighbour node graph
```

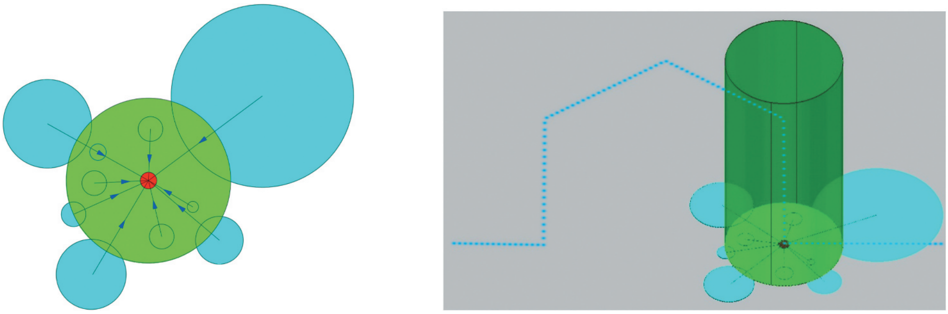


Figure 7. An example of the proposed cylinder neighbour node graph.

Figure 7 shows the proposed cylinder neighbour node graph with edges representing the adjacency relationships between nodes. A node (red) is connected to its neighbour node (cyan) by edges (light blue), where light blue dotted lines represent the building, and the red node and the cyan node are the centre node and neighbour nodes of the centre node, respectively. The circle size in the figure represents the node size (the number of the object point clouds), where the larger the circle size is, the more the number of point clouds of the node. The green circle is the buffer zone of the node object, and cylinder neighbour node graph is constructed between red node and light blue node.

## 3.4. *Instance segmentation*

In this article, we propose an energy minimization-based object merging method to achieve building instance segmentation by cylinder neighbour node relationship graph between objects.

### 3.4.1. *Merging of roof plane objects*

For the same building instance, the distance between the roof plane objects and the model consistency difference are the smallest, and the model consistency is the strongest (Roof plane objects conforms to model consistent whether they are gable roofs or flat roofs). Roof plane objects are regarded as centre nodes to construct energy function. The energy function Equation (10) is composed of Equation (11), Equation (13) and Equation (15), and it is used to calculate the minimum energy within cylinder neighbour radius of centre node. According to the minimum energy value between objects, node features are merged and updated.

### 3.4.2. *Merging of facade objects and roof accessory objects*

Since facade objects are below roof objects, and roof accessory objects are above roof plane objects. Facade object and roof accessory objects belong to the same building instance. Therefore, the two-dimensional distance between them and the roof instance is relatively close, the model consistency is the strongest, and the model consistency difference is the smallest, as shown in Figure 8. The two-dimensional Euclidean distance between the purple and cyan facades and the green roof instance is smaller than that of the yellow roof instance in the neighbour, and the consistency of the column model is larger than that of the yellow roof instance. The consistency value of the column model for the purple and cyan facade objects is close to 1, respectively. The consistency value of the column model of the green roof instance is also close to 1. After merging the purple and cyan facade objects with the green roof instance, the consistency difference between the column model and the green roof instance is smaller than that of the yellow roof instance. The facade objects and the roof accessory objects are regarded as centre node to construct energy function. The energy function Equation (10) is used to calculate the minimum energy between centre node and roof plane object within cylinder neighbour radius of centre node, where Equation (10) is composed of Equation (12), Equation (14) and Equation (16). The node is merged into the roof plane object according to the minimum energy, and the node feature is updated.

### *3.4.3. Energy function construction*

According to node feature, graph cut multi-label optimization method is used to optimize objects merging, which aims to eliminate and reduce false instance segmentation, and improve the topology consistency of building instance segmentation.

It is assumed that the building instance model is composed of column structures, and object feature merging task can be regarded as a labelling problem and formulated in terms of energy minimization as Equation (10) (Delong et al. 2010; Isack and Boykov 2012; Yan, Jie, and Jiang 2014)

$$E(f) = D_{data}\left(f_{pq}\right) + w_q\left(f_q\right) + h_{pq}\left(f_{pq}\right) \tag{10}$$

where $p$ is a centre node, $q$ is a node within the cylinder neighbour radius of the centre node $p$ that is with more points than the central node $p$. $N$ means node collection. $f_p \neq f_q$ means that node $p$ and node $q$ are not the same node. The data cost item (the first item in Equation (10)) measures the possibility of two nodes being merged, and the smaller the two-dimensional Euclidean distance between two nodes, the greater the probability that

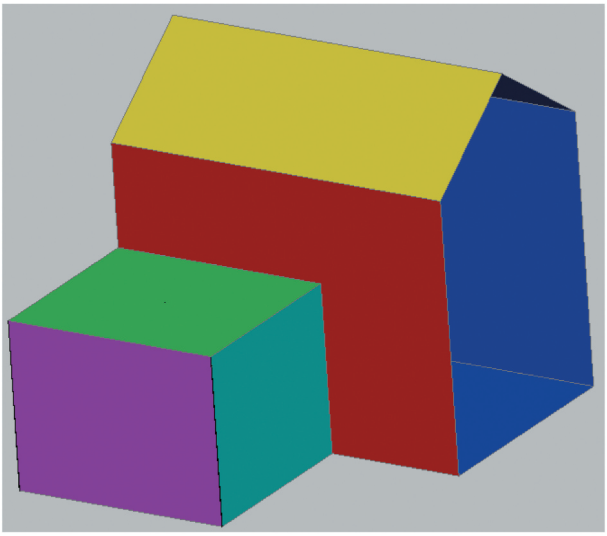


**Figure 8.** Diagram of object merging and updating.

the two nodes will be marked as the same building instance. The smooth cost item (the second item in Equation (10)) measures node model consistency, and the stronger the model consistency of this node is, the greater the probability of the object merged and updated is. The label cost item (the third item in Equation (10)) measures objects model consistency difference between nodes, and the smaller the nodes model consistency difference value between nodes is, the greater the probability of the node merged is.

The data cost term $D_{\mathrm{data}}(f_{pq})$ penalizes the difference of two-dimensional Euclidean distance between nodes. The two-dimensional Euclidean distance between nodes includes two-dimensional Euclidean distance between roof plane objects, two-dimensional Euclidean distance between roof plane objects and facade objects and two-dimensional Euclidean distance between roof plane objects and roof accessory objects, which are very small in the same building instance. According to two-dimensional Euclidean distance between nodes, the data cost term $D_{\mathrm{data}}(f_{pq})_{roof}$ of two-dimensional Euclidean distance between roof plane objects is calculated as Equation (11). The data cost term $D_{\mathrm{data}}(f_{pq})_{facade}$ of two-dimensional Euclidean distance between roof plane objects and facade objects, and two-dimensional Euclidean distance between roof plane objects and roof accessory objects is calculated as Equation (12).

$$D_{\mathrm{data}}(f_{pq})_{roof} = \begin{cases} exp^{dis_{pqmin}} & (dis_{pqmin} < dis2d) \\ +\infty & (dis_{pqmin} > dis2d) \end{cases} \tag{11}$$

$$D_{\mathrm{data}}(f_{pq})_{facade} = exp^{dis_{pqmin}} \tag{12}$$

where $dis_{pqmin}$ is two-dimensional Euclidean distance between node p and node q in object features extraction section.

The smooth cost term $w_q(f_q)$ penalizes model consistency of the same building instance. The building instance is generally composed of one or more objects. The model consistency value of each object is close to 1, and the model consistency value of the same building instance is also close to 1. In this paper, model consistency value $rt_c$ (Zhang et al. 2021) of objects is calculated. According to model consistency value, the smooth cost term $w_{smooth}(f_q)_{roof}$ of roof plane objects is calculated as Equation (13), and the smooth cost term $w_{smooth}(f_q)_{facade}$ of facade objects and roof accessory objects is calculated as Equation (14).

$$w_{smooth}(f_q)_{roof} = \begin{cases} \alpha * exp^{rt_c} & (rt_c \leq rt_t) \\ +\infty & (rt_c > rt_t) \end{cases} \tag{13}$$

$$w_{smooth}(f_q)_{facade} = \alpha * exp^{rt_c} \tag{14}$$

where $\alpha$ is a constant factor, which is used to adjust the weight among data cost item, smoothing cost item and label cost items. $rt_t$ is model consistency threshold value of node, and the centre node is marked as a non-merged node if $rt_c$of the node is less than the node model consistency threshold $rt_t$. Otherwise, the node will be merged.

The label cost item $h_{pq}(f_{pq})$ is used to punish the label inconsistency between node. For the same building instance, the model consistency value is the strongest, and the model consistency difference value $df$ between nodes is the smallest. In order to assign different nodes of the same instance to the same label as much as possible, the label cost item

$h_{label}\left(f_{pq}\right)_{roof}$ of roof plane objects is calculated as Equation (15). The label cost item $h_{label}\left(f_{pq}\right)_{facade}$ of building facade objects and roof accessory objects is calculated as Equation (16)

$$h_{label}\left(f_{pq}\right)_{roof} = \begin{cases} \beta * exp^{df} & (df \leq df_t) \\ +\infty & (df > df_t) \end{cases} \tag{15}$$

$$h_{label}\left(f_{pq}\right)_{facade} = \beta * exp^{df} \tag{16}$$

where $\beta$ is a constant factor, which is used to adjust the weight among data items, smoothing items, and label items. $df_t$ is model consistency difference threshold value as Equation (9) in object feature extraction section. The centre node will not be merged if model consistency difference between centre node p and node $q$ (node $q$ within cylinder neighbour radius of the centre node is with more points than the centre node $p$) is greater than $df_t$. Otherwise, the node will be merged.

### 3.4.4. Merging of isolated building instance

Threshold $num_3$ of the number of the building instance points is used to judge whether each building instance is an isolated building instance. The building instance is an isolated building instance if the number of the building instance points is less than the threshold num3, and then each point of the isolated building instance is regarded as search centre to find the closest building instance of the point. Next, the label of the building and the distance between the point to the closest building and the number of closest building instances in each cluster is recorded. The discrete point cloud cluster is merged into the closest building instance if there is only one. Otherwise, the minimum distance between the point of the cluster and the corresponding buildings is found and the point cloud clusters are merged to the corresponding building instance.

## 4. Results and analysis

### 4.1. Datasets description

The Dutch data set AHN3 (Actueel Hoogtebestand Nederland) is utilized to verify our proposed OBBIS method. For the acquisition of the AHN3 data set, various companies have used mostly the Riegl LMS-Q680i laser scanning sensor and sometimes the Riegl VQ-780i (Varney, Asari, and Graehling 2020; Du et al. 2019). The flight altitudes of the data set are between 450 and 500 m. The average point spacing is roughly 0.25 m (Du et al. 2019). The LiDAR dataset used in this paper was a published benchmark dataset, which was classified as buildings. The AHN3 data set is freely available to the public.[1] We used a partial scene of the data set, as shown in Figure 9.

### 4.2. Evaluation criteria

The performance of our proposed OBBIS method is assessed by object-based evaluation method and point-based evaluation method. The mutual overlap is calculated by *IoU* (Wu et al. 2020) to evaluate completeness, correctness, mean accuracy of building instance segmentation and loss rate of point clouds (Liang et al. 2018). When the *IoU* is larger than the threshold of minimum overlap, the building instance is identified as a correctly segmented building instance. The building instance is an over-segmentation building instance if the *IoU* is less than the threshold of the minimum overlap and the point clouds corresponding to the automatically segmented building instance contained only one building instance point clouds in the ground truth. On the other hand, the building instance is an under-segmentation building instance if the *IoU* is less than the threshold of the minimum overlap and the point cloud corresponding to the automatically segmented building instance contained multiple building instance point clouds in the ground truth. The *IoU* is used to evaluate building instance segmentation accuracy (Wu et al. 2020) as Equation (17).

$$IoU = \frac{num_{apoint} \cap num_{gpoint}}{num_{apoint} \cup num_{gpoint}} \tag{17}$$

where $num_{apoint}$ represents the number of automatically segmented building instance points, and$num_{gpoint}$ represents the number of building instance points in the ground truth. Object based building instance segmentation completeness ($com_{obj}$) and point based building instance segmentation completeness ($com_{point}$) are defined as the number of correctly segmented building instances and the number of correctly segmented building instance points with respect to the number of building instances and the number of building instance points in the ground truth, respectively, as Equation (18) and (19)

$$com_{obj} = \frac{num_{cobj}}{num_{gobj}} \tag{18}$$

$$com_{point} = \frac{num_{cpoint}}{num_{gpoint}} \tag{19}$$

where $num_{cobj}$ and $num_{gobj}$ represent the number of correctly segmented building instances and the number of building instances in the ground truth, respectively. $num_{cpoint}$represents the number of correctly segmented building instance point. Object based building instance segmentation correctness ($cor_{obj}$) and point based building instance segmentation correctness ($cor_{point}$) are defined as the number of correctly segmented building instances and the number of correctly segmented building instance

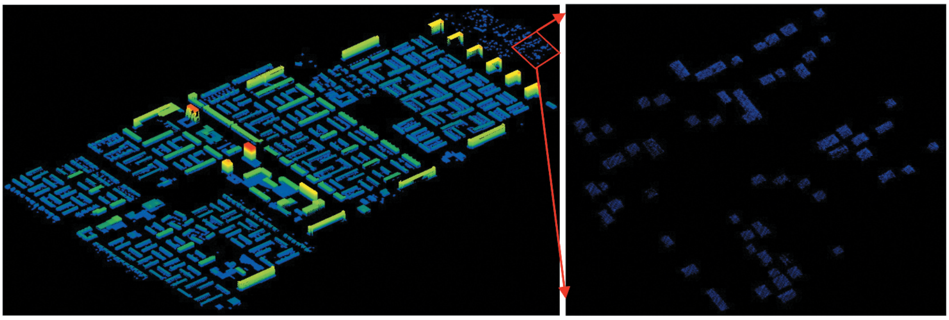


**Figure 9.** Building point clouds.

points with respect to the number of automatically segmented building instances and the number of automatically segmented building instance points, respectively, as Equation (20) and (21).

$$cor_{obj} = \frac{num_{cobj}}{num_{aobj}} \tag{20}$$

$$cor_{point} = \frac{num_{cpoint}}{num_{apoint}} \tag{21}$$

where $num_{aobj}$ represents the number of automatically segmented building instance. The mean accuracy ($ma_{obj}$) of object based building instance segmentation is defined as the joint probability that automatically segmented building instance randomly chosen was a correct segmentation and that ground truth building instance randomly chosen is automatically segmented building instance by proposed OBBIS method, as Equation (21). The mean accuracy (($ma_{point}$) of point based building instance segmentation is defined as the joint probability that automatically segmented building instance point randomly chosen is a correct segmentation building instance point and that ground truth building instance point randomly chosen is automatically segmented building instance point by proposed OBBIS method, as Equation (22).

$$\mathrm{ma}_{obj} = \frac{2*num_{cobj}}{num_{aobj} + num_{gobj}} \tag{22}$$

$$\mathrm{ma}_{point} = \frac{2*num_{cpoint}}{num_{apoint} + num_{gpoint}} \tag{23}$$

Loss rate of point clouds ($LR_{point}$) is defined as the percentage of miss building points to the building points in the ground truth, as Equation (24).

$$LR_{point} = \frac{num_{mpoint}}{num_{gpoint}} \tag{24}$$

where $num_{mpoint}$ is the number of loss points in the process of building instance segmentation.

### 4.3. Experimental results

The proposed OBBIS method is implemented in C++. The experiments were conducted on a computer with 8 GB RAM and an Intel Core i7-9750 H @ 2.59 GHz CPU. Figure 10 shows our experimental results for the proposed OBBIS method for complex buildings scenes, each building instance is represented by a colour. Because the selected small building point clouds contains a large scene, and the whole dataset covers an area of about 2 km$^2$. The scene contains a variety of complex small buildings, which poses a great challenge to the segmentation of individual buildings, but the result of the proposed OBBIS method building instance segmentation is good. The typical areas are circled with different color rectangle in the big building scenes and shown in Figure 11. Figure 11, (a)-(e) and Figure 11(f) are the results of several typical building instances segmentation in the scene of multiple connected small buildings. is the building scenes where the building spacing is much larger than the point cloud point spacing. is the big building scenes. From each typical scene, the algorithm in this paper has good segmentation results for small building instances.

To evaluate the performance of the proposed OBBIS method, building instance segmentation results were compared to the manually-marked ground truth in terms of its completeness, correctness, mean accuracy and loss rate of point clouds. Segmented building instance is considered correct when there is a minimum overlap of 80% with the corresponding building instance in the ground truth. The metrics in Table 1 show that the proposed OBBIS method performed well for complex scenes building instance segmentation of airborne LiDAR point clouds 94.16% ($com_{point}$), 78.92% ($com_{obj}$), 94.80% ($cor_{point}$), 87.07% ($cor_{obj}$), 97.33% ($ma_{point}$), 93.09% ($ma_{obj}$) and 0% ($LR_{point}$). The scene contains 3.58 million building points, and the running time of OBBIS is 573 seconds. Although the complex buildings scenes contain a large number of complex buildings $com_{obj}$, $com_{point}$, $cor_{obj}$, $cor_{point}$, $ma_{obj}$ and $ma_{point}$ of the OBBIS method are all above 78%.

### 4.4. Performance comparison

The performance of the proposed OBBIS method was further compared to the following eight benchmark methods. The eight benchmark methods includes the traditional connected-component labelling (TCCL) algorithm (Sohn and Dowman, 2007), the moving window (MW) method (Mohammad and Clive 2014), the voxel method (Lehtomaki et al. 2016), the region-growing-with-smoothness-constraint method (RGSC) (Axel and Aardt

Table 1. Quantitative evaluation of the proposed OBBIS method building instance segmentation result.

| Method | $com_{point}$ (%) | $com_{obj}$ (%) | $cor_{point}$ (%) | $cor_{obj}$ (%) | $ma_{point}$ (%) | $ma_{obj}$ (%) | $LR_{point}$ (%) | Runtime(s) |
|---|---|---|---|---|---|---|---|---|
| OBBIS | 94.16 | 78.92 | 94.80 | 87.07 | 97.33 | 93.09 | 0.00 | 573.15 |

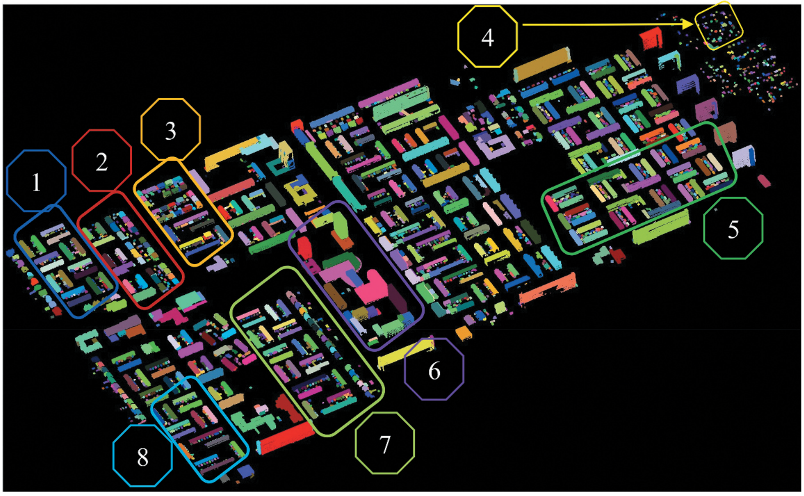
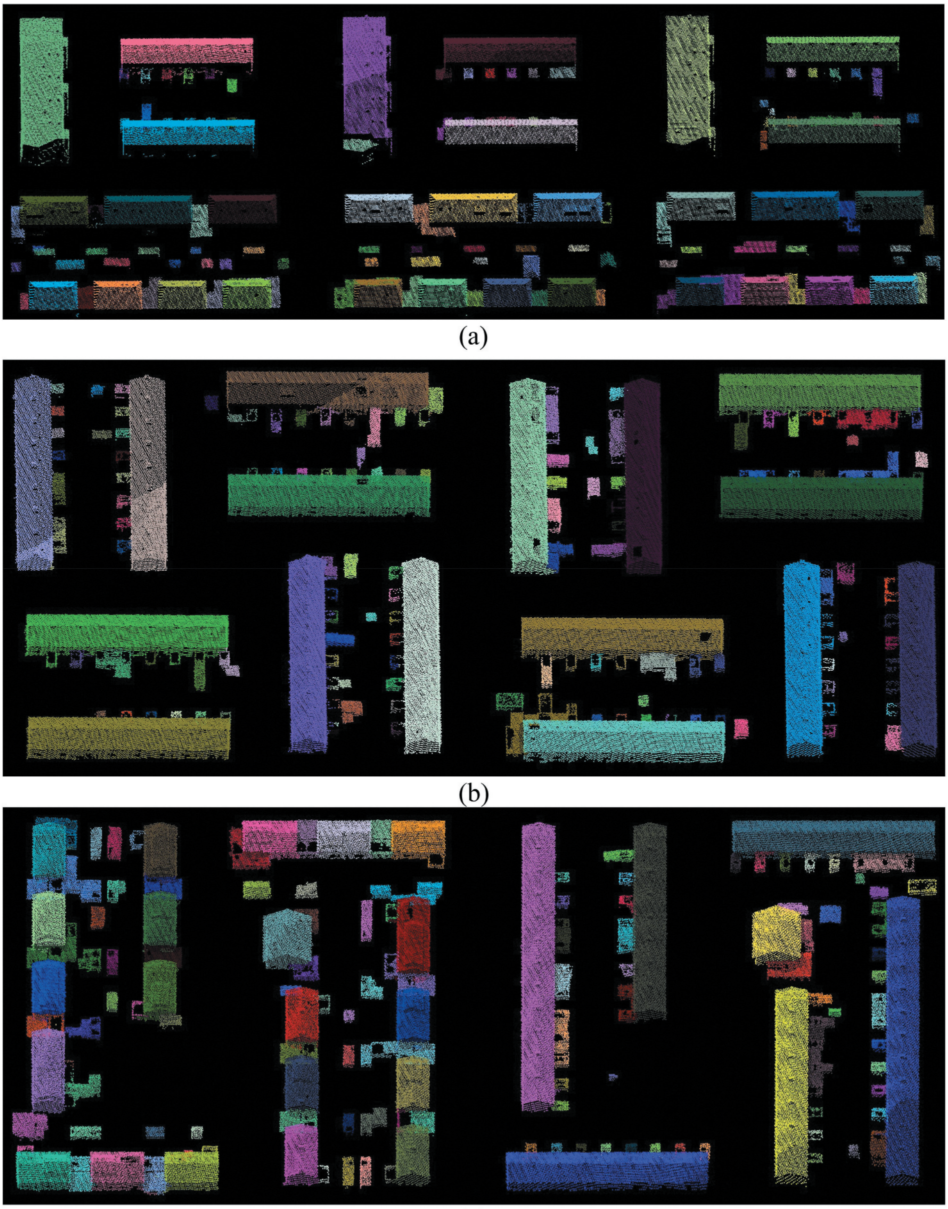


Figure 10. The proposed OBBIS method building instance result for various scenes.

2017), the two-dimensional DBSCAN segmentation method (DBSCAN2D) (Huang, Cao, and Cao 2018), the DBSCAN-clustering-with-normal-vector-constraint method (DBSCANNVC) (Ural and Shan 2020), the building instance segmentation method of combining RANSAC and DBSCAN (RANSAC_DBSCAN) (Wang et al. 2020b) and Euclidean clustering method (Gamal et al. 2020).

We conducted building instance segmentation using eight benchmark methods on the selected point clouds datasets. Table 2 lists the nine methods corresponding metrics on the

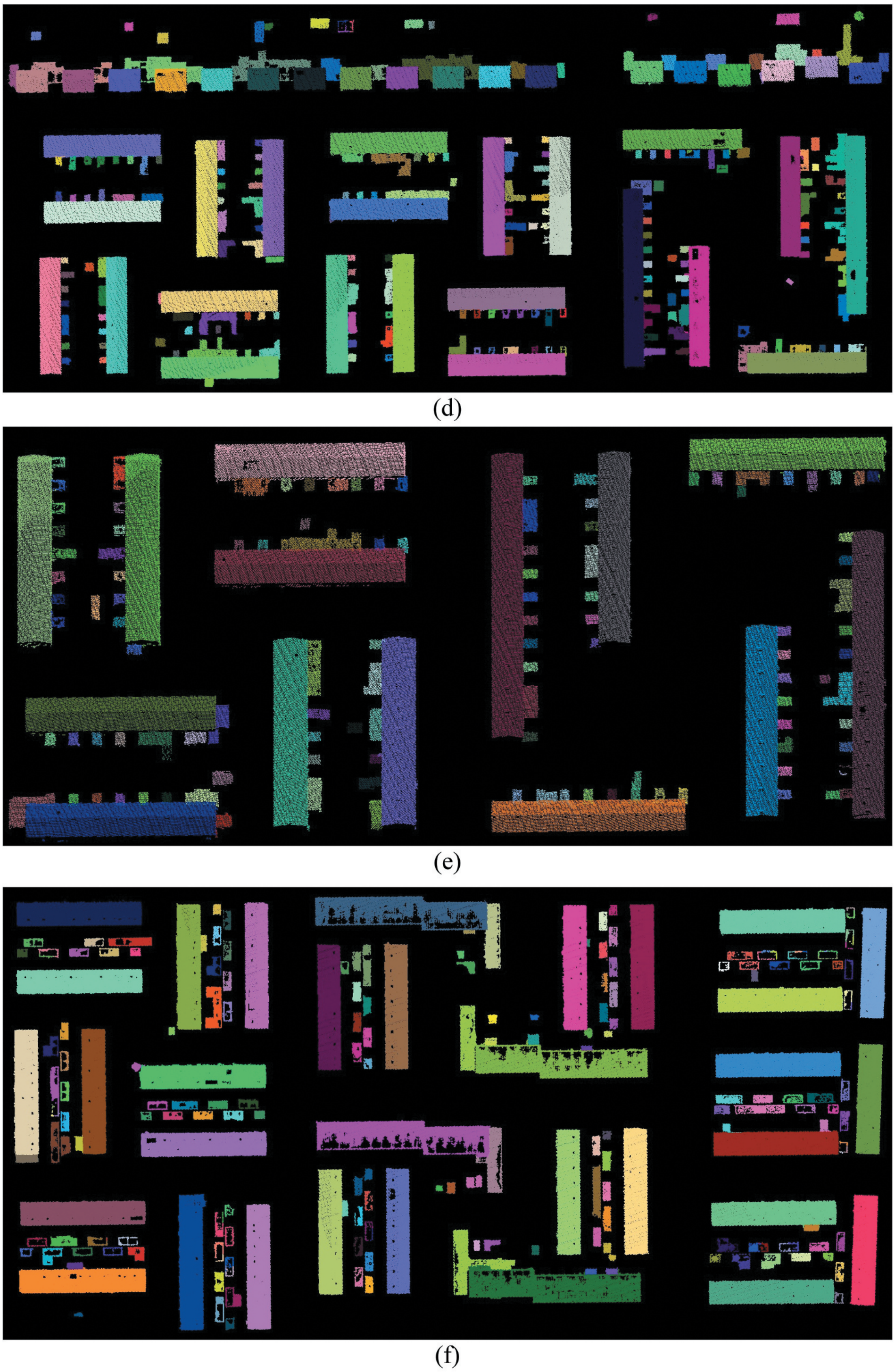


(a)

(b)

(c)

Figure 11. The local results of the proposed OBBIS method.

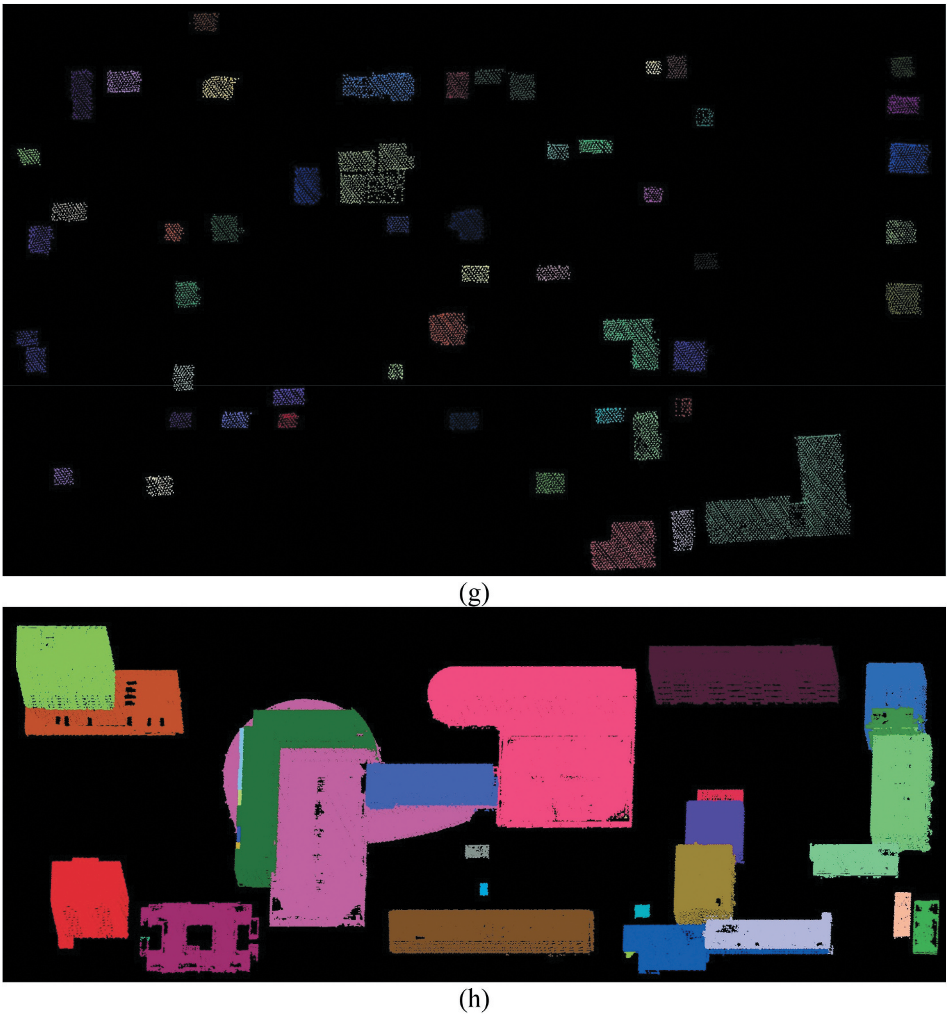


(d)

(e)

(f)

Figure 11. (Continued).



(g)

(h)

**Figure 11.** (Continued).

selected point clouds datasets, and $IoU_{0.80}$ was used to evaluate building instance segmentation (that is, $IoU_{0.80}$ was the minimum overlap of 80% with the corresponding building instance in the ground truth) (Dong et al. 2018). We found that the OBBIS method outperformed the benchmark methods on the selected point clouds, and more specifically, the following observations were made based on this comparison. (1) $com_{obj}$, $com_{point}$, $cor_{obj}$, $cor_{point}$, $ma_{obj}$ and $ma_{point}$ of eight benchmark methods are lower than the OBBIS method, as shown in Table 2. (2) $com_{obj}$, $cor_{obj}$ and $ma_{obj}$ of the voxel method, the RANSAC_DBScan3D method and the Euclidean Clustering method is lower than the OBBIS method and $com_{point}$, $cor_{point}$ and $ma_{point}$of the voxel method, the RANSAC_DBScan3D method and the Euclidean Clustering method is far lower than the OBBIS method, which means that building instance segmentation results of the voxel method, the RANSAC_DBScan3D method and the Euclidean Clustering method contain

Table 2. Performance comparison of building instance segmentation result.

| Method | $com_{point}$ (%) | $com_{obj}$ (%) | $cor_{point}$ (%) | $cor_{obj}$ (%) | $ma_{point}$ (%) | $ma_{obj}$ (%) | $LR_{point}$ (%) | Runtime(s) |
|---|---|---|---|---|---|---|---|---|
| TCCL | 69.13 | 69.80 | 72.08 | 22.93 | 83.78 | 37.31 | 4.09 | **5.13** |
| MW | 67.87 | 42.28 | 67.87 | 60.54 | 80.86 | 75.42 | **0.00** | 10.95 |
| voxel | 70.66 | 67.97 | 1.81 | 68.44 | 3.54 | 68.21 | **0.00** | 2512.60 |
| RGSC | 14.00 | 6.61 | 2.37 | 9.50 | 4.06 | 7.80 | 30.44 | 22.543 |
| DBSCAN2D | 67.17 | 40.69 | 67.20 | 84.05 | 80.38 | 91.33 | 0.05 | 56.34 |
| DBSCANNVC | 12.21 | 25.41 | 5.09 | 55.35 | 7.18 | 34.83 | 54.08 | 131.22 |
| RANSAC_DBScan3D | 67.24 | 69.34 | 19.93 | 72.64 | 30.74 | 70.96 | 4.54 | 2612.11 |
| Euclidean Clustering | 52.38 | 74.54 | 78.91 | 77.42 | 62.96 | 75.95 | 3.72 | 17.43 |
| OBBIS | **94.16** | **78.92** | **94.80** | **87.07** | **97.33** | **93.09** | **0.00** | 573.15 |

a large number of over-segmentation building instances. $com_{point}$, $cor_{point}$ and $ma_{point}$ of the TCCL method, the MW method, the DBSCAN2D method is lower than the OBBIS method, the $com_{obj}$, $cor_{obj}$ and $ma_{obj}$ of the TCCL method, the MW method, the DBSCAN2D method are far lower than the OBBIS method, which means that building instance segmentation results of the TCCL method, the MW method, the DBSCAN2D method contain a large number of under-segmented building instances. (3) $LR_{point}$ of the TCCL method, the RGSC method, the DBSCANNVC method, the RANSAC_DBScan3D method and the Euclidean Clustering method are 4.09%, 30.44%, 54.08%, 4.54% and 3.72%, respectively, which means that building instance segmentation results lost a large number point clouds, where the segmentation missing point clouds are mainly building façade point clouds. (4) Although the running time of the method proposed in this paper is not the shortest, it has run for less than ten minutes in 2457 building instances, fully meeting the application requirements. This paper focuses on improving the accuracy of the algorithm, and does not focus on the running time of the algorithm. Next, we will consider GPU parallel computing to solve the problem of the running efficiency of the proposed method. (5) In complex large scene, the experimental results of the OBBIS method are significantly better than the results of the eight benchmark methods, which means that the OBBIS method performed significantly better than the other methods for building instance segmentation of complex building scenes.

## 5. Conclusions

We proposed a novel airborne LiDAR point cloud building instance segmentation method called OBBIS. Based on the advantages of object segmentation, our OBBIS method combines each object using the relationship between neighbour objects to realize building instance segmentation. Experimental results demonstrated that the OBBIS method performed well on selected datasets, and our OBBIS method outperformed the other eight classical approaches. The completeness, correctness, mean accuracy of building instance segmentation are all above 78%. However, it is also demonstrated that the OBBIS method has limitations in complex scenes comprised of various building types, for example, when the building roof point cloud density was uneven, over-segmentation building instances occurred. In future work, we intend to improve the robustness of the OBBIS method to address these limitations.

## Note

[1] https://www.pdok.nl/nl/ahn3-downloads.

## Acknowledgements

This research was funded by the National Natural Science Foundation of China under Grant No. 42201474 and 41871368. The authors are grateful to the Dutch for AHN3 (Actueel Hoogtebestand Nederland) data set.

## Disclosure statement

No potential conflict of interest was reported by the author(s).

## Funding

The work was supported by the National Natural Science Foundation of China [42201474, 41871368]

## ORCID

Xinyi Liu http://orcid.org/0000-0001-5333-8054

## References

Axel, C., and J. Aardt. 2017. “Building Damage Assessment Using Airborne Lidar.” *Journal of Applied Remote Sensing* 11 (04): 1. doi:10.1117/1.JRS.11.046024.

Ben-Shabat, Y., T. Avraham, M. Lindenbaum, and A. Fischer. 2017. “Graph Based Over-Segmentation Methods for 3D Point Clouds.“ *Computer Vision & Image Understanding* 174: 12–23.

Cao, R., Y. Zhang, X. Liu, and Z. Zhao. 2017. “3d Building Roof Reconstruction from Airborne Lidar Point Clouds: A Framework Based on a Spatial Database.” *International Journal of Geographical Information Science* 31 (7–8): 1359–1380. doi:10.1080/13658816.2017.1301456.

Claudia, P., Davide, V., and Lorenzo, B. 2016. “A Hierarchical Approach to Three-Dimensional Segmentation of LiDar Data at Single-Tree Level in a Multilayered Forest.“ *IEEE Transactions on Geoscience and Remote Sensing* 54(7): 4190–4203. doi:10.1109/tgrs.2016.2538203.

Delong, A., A. Osokin, H. N. Isack, and Y. Boykov. 2010. “Fast Approximate Energy Minimization with Label Costs.“ Computer Vision & Pattern Recognition. IEEE. doi:10.1109/cvpr.2010.5539897.

Delsolneux, A., L. Moisan, and J. M. Morel. 2008. “*From Gestalt Theory to Image Analysis: A Probabilistic Approach*.“ New York: Springer.

Deshapriya, N. L., M. N. Dailey, M. K. Hazarika, and H. Miyazaki , 2020.“ Vec2instance: Parameterization for Deep Instance Segmentation.“

Dong, Z., B. Yang, P. Hu, and S. Scherer . 2018. “An Efficient Global Energy Optimization Approach for Robust 3d Plane Segmentation of Point Clouds.” *ISPRS Journal of Photogrammetry Remote Sens* 137: 112–133. doi:10.1016/j.isprsjprs.2018.01.013.

Du, J., D. Chen, R. Wang, J. Peethambaran, P. T. Mathiopoulos, L. Xie, and T. Yun, 2019. “A Novel Framework for 2.5-D Building Contouring from Large-Scale Residential Scenes.“ *IEEE Transactions on Geoscience and Remote Sensing* 57 (6): 4121–4145. doi:10.1109/tgrs.2019.2901539.

Filin, S., and N. Pfeifer. 2005. “Neighborhood Systems for Airborne Laser Data.” *Photogrammetric Engineering and Remote Sensing* 71 (6): 743–755. doi:10.14358/pers.71.6.743.

Friedman, J. H., J. L. Bentley, and R. A. Finkel. 1977. “An Algorithm for Finding Best Matches in Logarithmic Expected Time.Acm Trans.” *Mathematical Software* 3 (3): 209–226. doi:10.1145/355744.355745.

Gamal, A., A. Wibisono, S. B. Wicaksono, M. A. Abyan, N. Hamid, H. A. Wisesa, W. Jatmiko, and R. Ardhianto . 2020. “Automatic LIDAR Building Segmentation Based on DGCNN and Euclidean Clustering.” *Journal of Big Data* 7: 1–18. doi:10.1186/s40537-020-00374-x.

Grigorijs, G., M. Stefan, L. Shaun, and Andrew, E. 2018. ”Efficiency of Individual Tree Detection Approaches Based on Light-Weight and Low-Cost UAS Imagery in Australian Savannas.” *Remote Sensing* 10 (2): 161. doi:10.3390/rs10020161.

Haala, N., and M. Kada. 2010. “An Update on Automatic 3D Building Reconstruction.” *ISPRS Journal of Photogrammetry and Remote Sensing* 65 (6): 570–580. doi:10.1016/j.isprsjprs.2010.09.006.

Hackel, T., N. Savinov, L. Ladicky, Wegner, J., Schindler, K., and Pollefeys, M. 2017. ”Semantic3d.Net: A New Large-Scale Point Cloud Classification Benchmark.” *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*. doi:10.5194/isprs-annals-iv-1-w1-91-2017.

Hafiz, A. M., and G. M. Bhat. 2020. “A Survey on Instance Segmentation: State of the Art.” *International Journal of Multimedia Information Retrieval* 9 (3): 171–189. doi:10.1007/s13735-020-00195-x.

He, K., G. Gkioxari, P. Dollár, and R. Girshick , 2017. “Mask R-CNN.“ IEEE Transactions on Pattern Analysis & Machine Intelligence.

Huang, X., R. Cao, and Y. Cao. 2018. “A Density-Based Clustering Method for the Segmentation of Individual Buildings from Filtered Airborne LiDar Point Clouds.” *Journal of the Indian Society of Remote Sensing* 47 (6): 907–921. doi:10.1007/s12524-018-0911-y.

Iglovikov, V., S. Seferbekov, A. Buslaev, and A. Shvets, 2018. Ternausnetv2: Fully Convolutional Network for Instance Segmentation In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). doi:10.1109/cvprw.2018.00042.

Isack, H. N., and Y. Boykov. 2012. “Energy-Based Geometric Multi-Model Fitting.” *International Journal of Computer Vision* 97 (2): 123–147. doi:10.1007/s11263-011-0474-7.

Jing, L., B. Hu, J. Li, and Noland, T. 2012. ”Automated Delineation of Individual Tree Crowns from Lidar Data by Multi-Scale Analysis and Segmentation.” *Photogrammetric Engineering & Remote Sensing* 78 (12): 1275–1284. doi:10.14358/pers.78.11.1275.

Ji, S., Y. Shen, M. Lu, and M. Zhang. 2019a. “Building Instance Change Detection from Large-Scale Aerial Images Using Convolutional Neural Networks and Simulated Samples.” *Remote Sensing* 11 (11): 1343. doi:10.3390/rs11111343.

Ji, S., S. Wei, and M. Lu. 2019b. “Fully Convolutional Networks for Multisource Building Extraction from an Open Aerial and Satellite Imagery Data Set.” *IEEE Transactions on Geoscience and Remote Sensing* 57 (1): 574–586. doi:10.1109/tgrs.2018.2858817.

Jonathan, E., C. Roberts, S. Presentations, L. Linsen, and H. Prautzsch. 2001. “Local Versus Global Triangulations.“ Eurographics (Short Presentations).

Kang, X., J. Liu, and X. Lin. 2014. “Streaming Progressive TIN Densification Filter for Airborne LiDar Point Clouds Using Multi-Core Architectures.” *Remote Sensing* 6 (8): 7212–7232. doi:10.3390/rs6087212.

Kurdi, F. T., and M. Awrangjeb. 2020. “Automatic Evaluation and Improvement of Roof Segments for Modelling Missing Details Using Lidar Data.” *International Journal of Remote Sensing* 41 (12): 4702–4725. doi:10.1080/01431161.2020.1723180.

Lee, J., X. Cai, J. Lellmann, Dalponte, M., Malhi, Y., Butt, N., and Coomes, D. 2017. ”Individual Tree Species Classification from Airborne Multisensor Imagery Using Robust PCA.” *IEEE Journal of Selected Topics in Applied Earth Observations & Remote Sensing* 9 (6): 2554–2567. doi:10.1109/jstars.2016.2569408.

Lee, I., and T. Schenk. 2002. ”Perceptual Organization of 3D Surface Points.” *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 20-26 (34): 193–198. Part 3A.

Lee, H., K.C. Slatton, B.E. Roth, and Jr, W.P.C. 2010. ”Adaptive Clustering of Airborne LiDar Data to Segment Individual Tree Crowns in Managed Pine Forests.” *International Journal of Remote Sensing* 31 (1–2): 117–139. doi:10.1080/01431160902882561.

Lehtomaki, M., A. Jaakkola, J. Hyyppa, J. Lampinen, H. Kaartinen, A. Kukko, E. Puttonen, and H. Hyyppä. 2016. "Object Classification and Recognition from Mobile Laser Scanning Point Clouds in a Road Environment." *IEEE Transactions on Geoscience and Remote Sensing* 54 (2): 1226–1239. doi:10.1109/tgrs.2015.2476502.

Liang, X., H. Juha, K. Matti, P. Jiri, P. Norbert, H. Markus, B. Gábor, F. Pirotti, and H. Jan. 2018. "International Benchmarking of Terrestrial Laser Scanning Approaches for Forest Inventories." *ISPRS Journal of Photogrammetry Remote Sens* 144: 137–179. doi:10.1016/j.isprsjprs.2018.06.021.

Li, Q., L. Mou, Y. Hua, Y. Sun, P. Jin, Y. Shi, and X. Zhu, 2020. "Instance Segmentation of Buildings Using Keypoints." IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium. IEEE, 1452–1455. doi:10.1109/igarss39084.2020.9324457.

Lin, X., and J. Zhang. 2014. "Segmentation-Based Filtering of Airborne LiDar Point Clouds by Progressive Densification of Terrain Segments." *Remote Sensing* 6 (2): 1294–1326. doi:10.3390/rs6021294.

Liu, R. F., X. J. Ma, and X. S. Lu. 2022. "Automatic Extraction of Urban Road Boundaries Using Diverse LBP Features." *National Remote Sensing Bulletin* 26 (3): 14. doi:10.1016/j.jag.2017.09.010.

Liu, R., P. Wang, Z. Yan, X. Lu, M. Wang, J. Yu, M. Tian, and X. Ma. 2020. "Hierarchical Classification of Pole-like Objects in Mobile Laser Scanning Point Clouds." *The Photogrammetric Record* 35 (169): 81–107. doi:10.1111/phor.12307.

Liu, X., Y. Zhang, X. Ling, Y. Wan, and Q. Li. 2019. "Topolap: Topology Recovery for Building Reconstruction by Deducing the Relationships Between Linear and Planar Primitives." *Remote Sensing* 11 (11): 1372. doi:10.3390/rs11111372.

Mohammad, A., and F. Clive. 2014. "Automatic Segmentation of Raw LIDAR Data for Extraction of Building Roofs." *Remote Sensing* 6 (5): 3716–3751. doi:10.3390/rs6053716.

Moisan, E., C. Heinkele, P. Foucher, Charbonnier, P., Grussenmeyer, P., and Guillemin, S. 2021. "Combining Photogrammetric and Bathymetric Data to Build a 3D Model of a Canal Tunnel." *The Photogrammetric Record* 36 (175): 202–223. doi:10.1111/phor.12379.

Nahr, S. T., and M. Saadatseresht. 2021. "Utility-Pole Detection Based on Interwoven Column Generation from Terrestrial Mobile Laser Scanner Data." *The Photogrammetric Record* 36 (176): 402–424. doi:10.1111/phor.12394.

Papon, J., A. Abramov, M. Schoeler, and F. Worgotter. 2013. "Voxel Cloud Connectivity Segmentation - Supervoxels for Point Clouds." Computer Vision & Pattern Recognition. IEEE. 2027–2034. doi:10.1109/cvpr.2013.264.

Ramiya, A. M., R. R. Nidamanuri, and R. Krishnan. 2017. "Segmentation Based Building Detection Approach from LiDar Point Cloud." *Egyptian Journal of Remote Sensing and Space Science* 20 (1): 71–77. doi:10.1016/j.ejrs.2016.04.001.

Sithole, G., and G. Vosselman. 2004. "Experimental Comparison of Filter Algorithms for Bare-Earth Extraction from Airborne Laser Scanning Point Clouds." *Isprs Journal of Photogrammetry & Remote Sensing* 59 (1–2): 85–101. doi:10.1016/j.isprsjprs.2004.05.004.

Sohn G and Dowman I. 2007. "Data fusion of high-resolution satellite imagery and LiDAR data for automatic building extraction." *ISPRS Journal of Photogrammetry and Remote Sensing* 62(1): 43–63. doi: 10.1016/j.isprsjprs.2007.01.001.

Ural, S., and J. Shan. 2020. "Min-Cut Based Semantic Building Labeling for Airborne Lidar Data." *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 305–312. doi:10.5194/isprs-annals-v-2-2020-305-2020.

Uy, M. A., Q. H. Pham, B. S. Hua, Nguyen, T., and Yeung, S.K., 2019. "Revisiting Point Cloud Classification: A New Benchmark Dataset and Classification Model on Real-World Data." Proceedings of the IEEE/CVF international conference on computer vision, 1588–1597. doi:10.1109/iccv.2019.00167.

Varney, N., V. K. Asari, and Q. Graehling. 2020. "ALES: A large-scale aerial LiDAR data set for semantic segmentation." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops 186–187. doi:10.1109/cvprw50498.2020.00101.

Wang, X., T. O. Chan, K. Liu, W. Li, and C. Wei. 2020a. "Segmentation of Closely Packed Buildings from Airborne Lidar Point Clouds." The 11th International Conference on Mobile Mapping Technology.

Wang, X., T. O. Chan, K. Liu, J. Pan, M. Luo, W. Li, and C. Wei. 2020b. "A Robust Segmentation Framework for Closely Packed Buildings from Airborne LiDar Point Clouds." *International Journal of Remote Sensing* 41 (14): 5147–5165. doi:10.1080/01431161.2020.1727053.

Wang, J., R. Lindenbergh, and M. Menenti. 2018. "Scalable Individual Tree Delineation in 3D Point Clouds." *The Photogrammetric Record* 33 (163): 315–340. doi:10.1111/phor.12247.

Wang, M., and Y. Tseng. 2011. "Incremental Segmentation of Lidar Point Clouds with an Octree-Structured Voxel Space." *The Photogrammetric Record* 26 (133): 32–57. doi:10.1111/j.1477-9730.2011.00624.x.

Wang, H., W. Zhang, Y. Chen, M. Chen, and K. Yan. 2015. "Semantic Decomposition and Reconstruction of Compound Buildings with Symmetric Roofs from LiDar Data and Aerial Imagery." *Remote Sensing* 7 (10): 13945–13974. doi:10.3390/rs71013945.

Wang, L., Y. Zhao, and L. Yu. 2018. "A Greyscale Voxel Model for Airborne Lidar Data Applied to Building Detection." *The Photogrammetric Record* 33 (164): 470–490. doi:10.1111/phor.12266.

Weinmann, M., B. Jutzi, S. Hinz, and M. Clement. 2015. "Semantic Point Cloud Interpretation Based on Optimal Neighborhoods, Relevant Features and Efficient Classifiers." *ISPRS Journal of Photogrammetry and Remote Sensing* 105 : 286–304. doi:10.1016/j.isprsjprs.2015.01.016.

Wu, C., X. Hu, M. Happold, Q. Xu, and U. Neumann. 2020. "Geometry-Aware Instance Segmentation with Disparity Maps." doi:10.1109/ccdc52312.2021.9602282.

Xiao, Y., C. Wang, J. Li, Zhang, W., Xi, X., Wang, C., and Dong, P., 2015. "Building Segmentation and Modeling from Airborne LiDar Data." *International Journal of Digital Earth* 8 (9). doi:10.1080/17538947.2014.914252.

Xu, A. H., J. Z. Chen, S. F. Lu, R. H. Liang, and L. L. Nan, 2021. "3D Instance Segmentation of MVS Buildings." in Proceedings of the IEEE conference on computer vision and pattern recognition 60: 1–14. doi:10.1109/tgrs.2022.3183567.

Yang, W. S., L. L. Cai, and S. D. Gu. 2018. "Extraction City Road Boundary Method Based on Point Cloud Normal Vector Clustering." *Acta Photonica Sinica* 47 (6). doi:10.3788/gzxb20184706.0612003.

Yang, N., Z. Qin, Z. Li, and W. Xu. 2013. "Laser Footprint Detection and Separation Method of One Single Building Based on the Airborne LiDar Point Clouds." *Geomatics Science and Engineering* 33 (6): 4.

Yan, J., S. Jie, and W. Jiang. 2014. "A Global Optimization Approach to Roof Segmentation from Airborne Lidar Point Clouds." *Isprs Journal of Photogrammetry & Remote Sensing* 94 (august): 183–193. doi:10.1016/j.isprsjprs.2014.04.022.

Yan, L., and F. Wei. 2018. "Single Part of Building Extraction from Dense Matching Point Cloud." *Chinese Journal of Lasers*. 45 (07): 270–277. doi:10.3788/cjl201845.0710004.

Zeng, Q., J. Mao, X. Li, and Liu, X. 2007. "LIDAR Data Filtering and Classification with TIN and Assistant Plane." Proceedings of SPIE - The International Society for Optical Engineering, 6752:8–675206. doi:10.1117/12.760108.

Zhang, J., and X. Lin. 2013a. "Filtering Airborne LiDar Data by Embedding Smoothness-Constrained Segmentation in Progressive TIN den-Sification." *Isprs Journal of Photogrammetry & Remote Sensing*. 81: 44–59. doi:10.1016/j.isprsjprs.2013.04.001.

Zhang, J., X. Lin, and X. Liang. 2017. "Advances and Prospects of Information Extraction from Point Clouds." *Acta Geodaetica et Cartographica Sinica* 46 (10).

Zhang, J., X. Lin, and X. Ning. 2013b. "SVM-Based Classification of Segmented Airborne LiDar Point Clouds in Urban Areas." *Remote Sensing*. 5 (8): 3749–3775. doi:10.3390/rs5083749.

Zhang, Y., W. Yang, X. Liu, Y. Wan, and Y. Tan. 2021. "Unsupervised Building Instance Segmentation of Airborne LiDar Point Clouds for Parallel Reconstruction Analysis." *Remote Sensing*. 13 (6): 1136. doi:10.3390/rs13061136.

Zhao, Z., Y. Duan, Y. Zhang, and Cao, R. 2016a. "Extracting Buildings from and Regularizing Boundaries in Airborne Lidar Data Using Connected Operators." *International Journal of Remote Sensing* 37 (3–4): 889–912. doi:10.1080/01431161.2015.1137647.

Zhao, K., J. Kang, J. Jung, and G. Sohn, 2016b. "Building Extraction from Satellite Images Using Mask R-CNN with Building Boundary Regularization" Proceedings of the IEEE conference on computer vision and pattern recognition workshops 247–251. doi:10.1109/cvprw.2018.00045"