

Voxel-Based Extraction and Classification of 3-D Pole-Like Objects From Mobile LiDAR Point Cloud Data

Kang, Zhizhong; Yang, Juntao; Zhong, Ruofei; Wu, Yongxing; Shi, Zhenwei; Lindenberg, Roderik

DOI

[10.1109/JSTARS.2018.2869801](https://doi.org/10.1109/JSTARS.2018.2869801)

Publication date

2018

Document Version

Accepted author manuscript

Published in

IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing

Citation (APA)

Kang, Z., Yang, J., Zhong, R., Wu, Y., Shi, Z., & Lindenberg, R. (2018). Voxel-Based Extraction and Classification of 3-D Pole-Like Objects From Mobile LiDAR Point Cloud Data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(11), 4287-4298.
<https://doi.org/10.1109/JSTARS.2018.2869801>

Important note

To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

Voxel-Based Extraction and Classification of 3-D Pole-Like Objects From Mobile LiDAR Point Cloud Data

Zhizhong Kang , Juntao Yang , Ruofei Zhong, Yongxing Wu, Zhenwei Shi, and Roderik Lindenbergh

Abstract—The digital mapping of road environment is an important task for road infrastructure inventory and urban planning. Automatic extraction and classification of pole-like objects can remarkably reduce mapping cost and enhance work efficiency. Therefore, this paper proposes a voxel-based method that automatically extracts and classifies three-dimensional (3-D) pole-like objects by analyzing the spatial characteristics of objects. First, a voxel-based shape recognition is conducted to generate a set of pole-like object candidates. Second, according to their isolation and vertical continuity, the pole-like objects are detected and individualized using the proposed circular model with an adaptive radius and the vertical region growing algorithm. Finally, several semantic rules, consisting of shape features and spatial topological relationships, are derived for further classifying the extracted pole-like objects into four categories (i.e., lamp posts, utility poles, tree trunks, and others). The proposed method was evaluated using three datasets from mobile LiDAR point cloud data. The experimental results demonstrate that the proposed method efficiently extracted the pole-like objects from the three datasets, with extraction rates of 85.3%, 94.1%, and 92.3%. Moreover, the proposed method can achieve robust classification results, especially for classifying tree trunks.

Index Terms—Classification, detection, mobile LiDAR, principal component analysis (PCA), vertical pole-like objects.

I. INTRODUCTION

URBAN road environment contains a variety of pole-like objects such as street signs, lamp posts, utility poles, tree trunks, and so on. Mapping of these vertical pole-like objects is of significance for robot navigation, autonomous driving, detailed three-dimensional (3-D) map generation, road

infrastructure inventory, and monitoring [1]. For example, the positions of these pole-like objects might provide the navigation information needed for autonomous driving, especially in the area where global positioning system signal is weak. However, manual visual inspection is time-consuming and laborious since there are a large number of pole-like objects along road corridors. Thus, it is necessary to develop an automatic pole-like object detection and classification method to reduce manual inspection time and obtain the pole-like object information.

With the development of mobile mapping systems, many approaches for extracting pole-like objects from optical images or videos have been developed [2] and have the advantages of high-level automation. Nevertheless, these optical image based methods usually rely on the visibility of pole-like objects, which is easily affected by illumination changes and weather conditions (such as fog and rain) [3]. Compared with optical images or videos, mobile LiDAR data have become increasingly popular for automatically detecting pole-like objects since it can avoid several inevitable problems from which optical images suffer, such as variant illumination and background confusion, which can effectively reduce the error extraction rate.

To date, a large number of methods for pole-like object detection and classification from mobile LiDAR point cloud data have been presented. For pole-like object detection, most of the existing methods consist of two steps: segmentation and recognition. The segmentation step generates a set of connected clusters of points as object candidates, while the recognition step further distinguishes these object candidates into the pole-like objects or others. In the literature involving pole-like object detection, several segmentation methods, such as the region growing based methods [1], the clustering-based methods [4]–[6], the slicing-based methods [7], the density-based methods [8]–[11], the scan line based methods [12], [13], and the voxel-based methods [14]–[16], have been proposed and they all partitioned the original point cloud data into connected segments, each of which comprises the same type of points, for the subsequent recognition step. As a matter of fact, the integrated segmentation methods, such as combining the slicing-based method with the region growing based algorithm [1], combining the Euclidean distance clustering method with normalized cut segmentation method [4], as well as combining the scan line based segmentation method with the clustering method [12], were usually used to robustly obtain the accurate pole-like candidates, particularly

Manuscript received May 1, 2018; revised August 12, 2018; accepted September 4, 2018. This work was supported by the National Natural Science Foundation of China under Grant 41471360. (Corresponding author: Zhizhong Kang.)

Z. Kang, J. Yang, Y. Wu, and Z. Shi are with Beijing Advanced Innovation Center for Imaging Technology, Capital Normal University, Beijing 100048, China, and also with the School of Land Science and Technology, China University of Geoscience, Beijing 100083, China (e-mail: zzkang@cugb.edu.cn; jtyang66@126.com; 851449972@qq.com; szw_576487873@qq.com).

R. Zhong is with Beijing Advanced Innovation Center for Imaging Technology and Key Lab of 3D Information Acquisition and Application, College of Resource Environment and Tourism, Capital Normal University, Beijing 100048, China (e-mail: zrfss@163.com).

R. Lindenbergh is with the Department of Geoscience and Remote Sensing, Delft University of Technology, Delft 2629HS, The Netherlands (e-mail: r.c.lindenbergh@tudelft.nl).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JSTARS.2018.2869801

in overlapped regions. Then, the object-based geometric features were calculated for the representation of these segmented objects candidates that were further identified into pole-like objects or others using knowledge-based [5], [10], [12], [14], [15], [17], [18] or machine learning based approaches [1], [11]. In terms of these detection methods, the main challenge is how to robustly and efficiently segment and recognize the pole-like objects in the overlapped regions.

After detecting the individual pole-like objects, some research works also attempted to further classify these pole-like objects as lamp posts, street signs, utility poles, tree trunks, and so on. Yan *et al.* [5] calculated ensemble of shape functions and geometric features of the target objects and classified them using a random forest classifier. Rodríguez-Cuenca *et al.* [17] computed roughness and scattering degree and then labeled the vertical elements as man-made poles or trees. Yokoyama *et al.* [19] used shape features and context features for classifying utility poles, lamp posts, and street signs. Fukano and Masuda [20] subdivided a pole-like object into five point sets and then computed the feature vector for each subset, which served as the input of a random forest classifier. Riveiro *et al.* [21] extracted shape descriptors to classify retro-reflective traffic signs from mobile LiDAR data. Li *et al.* [22] decomposed the pole-like objects into different components according to their spatial relations and proposed a method to evaluate the decomposition results. However, due to the shape similarities among different types of pole-like objects, these existing classification methods also have some limitations of requiring a lot of training data or only using the shape features.

Aiming at addressing these challenges raised from the state-of-the-art of the pole-like object detection and classification, this paper proposes a voxel-based method for automatic extraction and classification of 3-D pole-like objects from mobile LiDAR point cloud data. The pole-like objects, especially in the overlapped regions, might not be segmented accurately. Instead of directly segmenting the individual and isolated point clusters for further identification, the voxel-based shape recognition is first conducted to generate a set of linear voxels as the pole-like object candidates based on the unique characteristics that the local parts of the pole-like objects are still pole-like. Second, according to their isolation and vertical continuity, the pole-like objects are detected and individualized using the proposed circular model with an adaptive radius and the vertical region growing algorithm. Third, due to the similar shape features among different types of pole-like objects, the spatial topological relationship is also introduced to assist in the pole-like object classification.

The rest of this paper is organized as follows. Section II describes our proposed method in detail. Section III presents experimental results and analysis for evaluating the proposed method. This paper concludes with a discussion of future research considerations in Section IV.

II. METHODOLOGY

In this paper, we detect and classify the pole-like objects from mobile LiDAR point cloud data through voxel-based analysis.

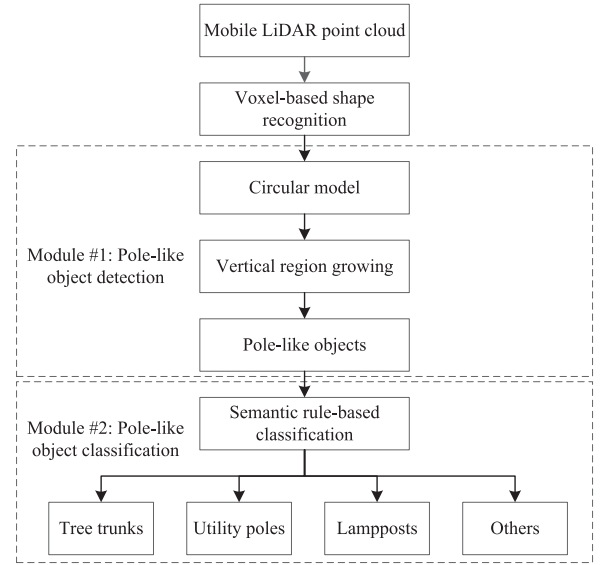


Fig. 1. Workflow of the proposed method.

As shown in Fig. 1, the proposed method consists of the following steps.

- 1) Generating a set of linear voxels as the pole-like object candidates through the voxel-based shape recognition.
- 2) Identifying voxels that belong to pole-like objects using the proposed circular model with an adaptive radius.
- 3) Extracting the individual pole-like objects based on the vertical region growing algorithm.
- 4) Further classifying the extracted pole-like objects by several semantic rules.

As a result, the pole-like objects are extracted and classified into four categories, i.e., tree trunks, utility poles, lamp posts, and others (e.g., street signs). Key algorithms of the proposed method are given in more detail below.

A. Voxel-Based Shape Recognition

The parts of the pole-like objects exhibit the linear pattern, whereas those of others (such as buildings and tree crowns) usually present the planar or spherical characteristics. Thus, in this section, we conduct the voxel-based shape recognition with a successive scheme, which includes voxelization, voxel-based dimension analysis, and MRF-based shape recognition optimization. As a result, a set of linear voxels is generated and serves as the input of the subsequent modules.

1) *Voxelization*: Mobile LiDAR point cloud data contain a large number of points and the distribution of these points is generally heterogeneous. Consequently, to reduce the data volume, we construct 3-D voxels as primitives based on the XYZ coordinates alone. The space is divided into a regular 3-D grid and each voxel is in the shape of a cuboid and its geometry is defined by length (l), width (w), and height (h). The location of a voxel is indexed by column (i), row (j), and layer (k). According to the minimum coordinate ($X_{\min}$, $Y_{\min}$, $Z_{\min}$), and the length (l), width (w), and height (h) of the 3-D voxel, the index (i, j, k) of each point in the point cloud can be calculated

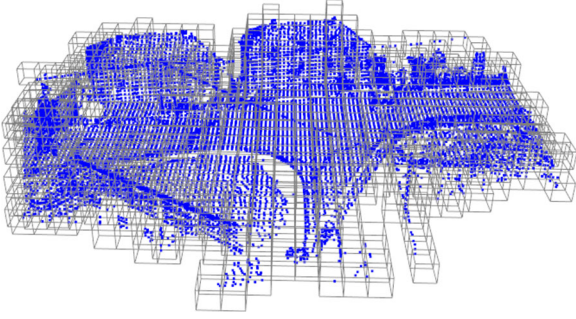


Fig. 2. Voxelization of the scattered point cloud.

using (1). As a result, numerous points are grouped together to form 3-D voxels. As shown in Fig. 2, a number of points are grouped together to form 3-D voxels:

$$\begin{aligned} i &= \text{floor}(X - X_{\min})/l \\ j &= \text{floor}(Y - Y_{\min})/w \\ k &= \text{floor}(Z - Z_{\min})/h. \end{aligned} \quad (1)$$

2) *Voxel-Based Dimension Analysis*: After voxelization, we use principal component analysis (PCA) [24] to analyze the dimension of voxels, since PCA is a well-accepted approach for dimension analysis and is widely used to infer the objects in point cloud data into three types of shape: linear, planar, and spherical [25].

Voxel dimensional analysis is a procedure that analyzes the local shape of points inside a voxel. It is well-known that the voxel size directly determines the number of points it contains, which probably affects the validity of the dimensional analysis. Alternatively, we carry out the dimensional analysis based on the geometric center p of the query voxel and its neighbors within a predefined radius R . For describing the geometric structure around p , the covariance matrix C_p can be derived from the query geometric center p and its neighbors using

$$C_p = \frac{1}{|N|} \sum_{p_i \in N} (p_i - \bar{p})(p_i - \bar{p})^T \quad (2)$$

where $|N|$ denotes the number of neighboring points, $\bar{p} = \frac{1}{|N|} \sum_{p_i \in N} p_i$. Let $\lambda_1 > \lambda_2 > \lambda_3 > 0$ be the normalized eigenvalues of C_p . Following the idea of the previous work [24], we identify the geometric structure of points using linear (a_{1d}), planar (a_{2d}), and spherical (a_{3d}) characteristics, shown as follows:

$$\left. \begin{aligned} a_{1d} &= \frac{\sqrt{\lambda_1} - \sqrt{\lambda_2}}{\sqrt{\lambda_1}} \\ a_{2d} &= \frac{\sqrt{\lambda_2} - \sqrt{\lambda_3}}{\sqrt{\lambda_1}} \\ a_{3d} &= \frac{\sqrt{\lambda_3}}{\sqrt{\lambda_1}} \end{aligned} \right\}. \quad (3)$$

However, the performance of the dimensional analysis is easily affected by the predefined radius R . For example, the geometric structure of points may be incorrectly estimated if the predefined radius R is too small, whereas the identification might be affected by noises when the predefined radius R is too large. Thus, in this paper, we adopt an entropy function [26] to adaptively determine the predefined radius R for inferring the

TABLE I
LOCAL SHAPE INFERENCE ACCORDING TO EIGENVALUES OF THE COVARIANCE MATRIX

Local shape	Values	Predominant direction
Linear	$a_{1d} > a_{2d}$ and $a_{1d} > a_{3d}$	v_1
Planar	$a_{2d} > a_{1d}$ and $a_{2d} > a_{3d}$	v_3
Spherical	$a_{3d} > a_{1d}$ and $a_{3d} > a_{2d}$	non-direction

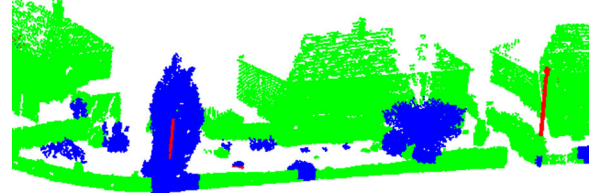


Fig. 3. Voxel-based shape recognition. Linear: red, Planar: green, and Spherical: blue.

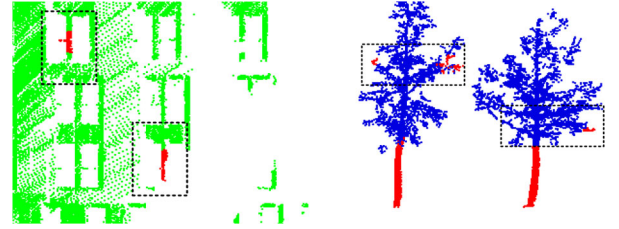


Fig. 4. Some typical illustrations of linear voxels that are not parts of pole-like objects. Linear: red, Planar: green, and Spherical: blue.

geometric structure of points, as follows:

$$E_f(V_p^R) = -a_{1d} \ln(-a_{1d}) - a_{2d} \ln(-a_{2d}) - a_{3d} \ln(-a_{3d}). \quad (4)$$

Let $R_{\min}$ and $R_{\max}$ be the minimum and maximum radius, respectively. The predefined radius R is iteratively increased by R_{add} for minimizing the entropy function $E_f(V_p^R)$. For the query geometric center p , the predefined radius R , the eigenvalues λ_1 , λ_2 , and λ_3 , and the corresponding eigenvectors v_1 , v_2 , and v_3 are stored. Then, a_{1d} , a_{2d} , and a_{3d} are calculated and compared (see Table I). When a_{1d} has the largest value, the point set within the voxel shows a linear shape and its predominant direction is the direction of the eigenvector v_1 , which is aligned with the direction of the linear object. When a_{2d} has the largest value, the point cloud within the voxel presents a planar shape and its predominant direction is the direction of the eigenvector v_3 , which is the normal direction of the plane. Finally, when a_{3d} has the largest value, the point cloud within the voxel exhibits a spherical shape and no predominant direction. Fig. 3 exhibits the voxel-based shape recognition results.

3) *MRF-Based Shape Recognition Optimization*: As shown in Fig. 4, the linear voxels identified in Section II-A2 are probably linear parts of tree branches or buildings rather than the pole-like objects because of the complex and incomplete scenes, occlusions, and noises. Furthermore, we also observe that unlike the pole-like objects, most parts of other ground objects, such as buildings and tree crowns, are planar and spherical. To address this issue in Fig. 4, an MRF model is commonly used to model

the contextual information for obtaining locally continuous and globally optimal results. Therefore, in this section, we aim to optimize the shape recognition results by introducing contextual information for the subsequent detection and classification modules and formulate the optimization problem into a binary labeling problem (i.e., voxels of linear and nonlinear type).

MRF model [27] is a weighted undirected graph $G = \langle V, E \rangle$, where V denotes a set of nodes corresponding to the voxels, and E represents a set of undirected edges between the neighboring nodes. In this paper, the weighted undirected graph also contains two additional terminals V_s and V_t , corresponding to linear and nonlinear type, respectively. For a point cloud D , let $\zeta = \{\zeta_1, \zeta_2, \dots, \zeta_N\}$ denote a set of voxels, let $\Omega = \{\text{linear}, \text{nonlinear}\}$ be a set of labels, and let $L = \{l = (l_{\zeta_1}, l_{\zeta_2}, \dots, l_{\zeta_N}) | l_{\zeta_i} \in \Omega, i = 1, 2, \dots, N\}$ denote all possible label configurations of voxels. As a result, the set of nodes $V = \{V_{\zeta_i} | i = 1 \dots N\} \cup \{V_s, V_t\}$ and the set of edges $E = \{\{V_u, V_v\}, \{V_u, V_s\}, \{V_u, V_t\}\}, u, v \in \{\zeta_i | i = 1 \dots N\}$. In the field of computer vision, finding the optimal label configurations L^* can be naturally formulated into the energy function minimization as follows:

$$\text{En}(L) = \text{En}_{\text{data}}(L) + \lambda \cdot \text{En}_{\text{smooth}}(L) \quad (5)$$

where data term $\text{En}_{\text{data}}(L)$ measures the disagreement between L and the observed data, smooth term $\text{En}_{\text{smooth}}(L)$ measures the extent to which L is not piecewise smooth, and λ denotes the weight parameter.

The form of data term $\text{En}_{\text{data}}(L)$ is typically defined as

$$\text{En}_{\text{data}}(L) = \sum_{u \in V} D_u(l_u) \quad (6)$$

where $D_u(l_u)$ quantitatively measures the degree of “fit” between the label l_u and the observed data, and is calculated using (7). The larger the value of a_{1d} , the smaller the data term $D_u(l_u)$:

$$D_u(l_u) = \begin{cases} 1 - e^{a_{1d}}, & \text{if } u \text{ is labeled as nonlinear} \\ e^{a_{1d}}, & \text{if } u \text{ is labeled as linear} \end{cases} \quad (7)$$

To generate locally continuous and globally optimal label configurations, the smooth term $\text{En}_{\text{smooth}}(L)$ is generally defined as

$$\text{En}_{\text{smooth}}(L) = \sum_{\{u,v\} \in \mathfrak{R}} S_{u,v}(l_u, l_v) \quad (8)$$

where $\mathfrak{R}$ denotes a 26 neighborhood system, $S_{u,v}(l_u, l_v) = g(u, v) \cdot \delta(l_u, l_v)$, $\delta(l_u, l_v) = \begin{cases} 1, & \text{if } l_u \neq l_v \\ 0, & \text{otherwise} \end{cases}$, $g(u, v) = e^{-\frac{D_{\zeta_u \zeta_v}}{\rho}}$, and $D_{\zeta_u \zeta_v}$ denotes the distance between ζ_u and ζ_v . ρ denotes the expectation of all neighboring distances [28]. As defined in (8), the smoothness penalty term is zero for the neighboring voxels with the same label. With regards to the adjacent voxels with different labels, the smaller the distance between them, the larger the smoothness penalty term. Consequently, the smooth term $\text{En}_{\text{smooth}}(L)$ encodes the extent to which the adjacent voxels belong to the same label. Table II lists the weight definition of edges in the weighted undirected graph. Finally, the energy function in (5) is minimized by the $\alpha - \beta$ swap algorithm [29], since it approximately minimizes the energy function for

TABLE II
WEIGHT DEFINITION OF EDGES

Edges	Weights
$\{V_s, V_u\}$	$D_u(s) + \sum_{v \in \{u\}} S_{u,v}(s, l_v), u \in \{\zeta_i i = 1 \dots N\}$
$\{V_t, V_u\}$	$D_u(t) + \sum_{v \in \{u\}} S_{u,v}(t, l_v), u \in \{\zeta_i i = 1 \dots N\}$
$\{V_u, V_v\}$	$S_{u,v}(l_u, l_v), u, v \in \{\zeta_i i = 1 \dots N\}$

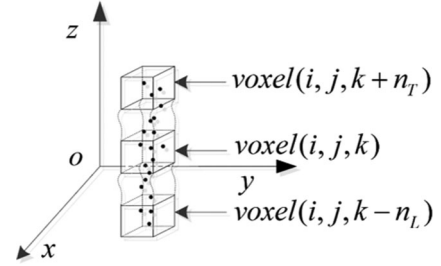


Fig. 5. Voxels of the pole-like objects. Each voxel can be indexed using (i, j, k) .

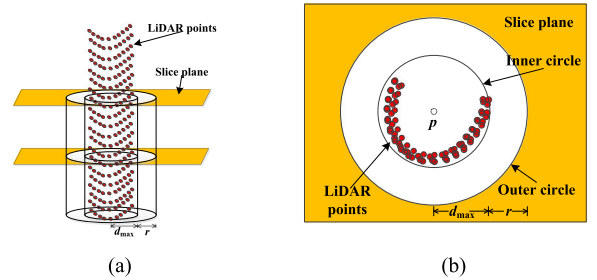


Fig. 6. Schematic diagram of the circular model with an adaptive radius. (a) Slice the point cloud and the circular model. (b) Detailed projective point distribution of a slice (partial).

an arbitrary finite set of labels, which not only demonstrates improved computational efficiency but also obtains the globally optimal solution.

B. Identifying Voxels of Pole-Like Objects Using a Circular Model With an Adaptive Radius

According to the procedure described in Section II-A, each voxel is labeled as linear, planar, or spherical type. It is noted that the pole-like objects are generally individual and isolated, and their predominant direction is approximately parallel to the Z-axis. As shown in Fig. 5, voxels within the single poles exhibit a linear arrangement in the vertical direction and no planar or spherical voxels exist in the horizontal direction. In contrast, other ground objects, such as tree canopies, low vegetations, and building facades, might have multiple continuous voxels in the horizontal direction and most of these voxels are planar or spherical. Hence, we follow the slicing strategy [7] and use a circular model with an adaptive radius [13] for identifying voxels that belong to the pole-like objects.

As illustrated in Fig. 6(a), we first divide the non-ground points into slices according to the selected interval (e.g., the

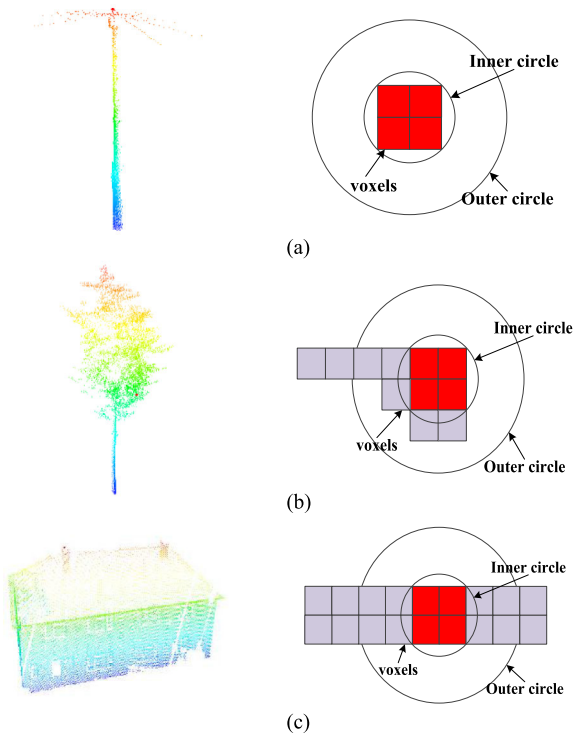


Fig. 7. Examples of detecting voxels of pole-like objects using the circular model with an adaptive radius. Point cloud data in the first row are rendered from blue to red as the height value increases. The red squares denote linear voxels, and the gray squares denote non-linear voxels. (a) Utility poles. (b) Street trees. (c) Buildings.

size of a voxel in this paper). After slicing, the linear voxels generated in Section II and its adjacent linear voxels within the same slice are clustered [30]. Then, the clustered points are projected onto the associated slice plane [see Fig. 6(b)]. As a matter of fact, the single pole-like objects should be narrow and elongated point clusters that are surrounded by empty space in the LiDAR point cloud data. Thus, we build a circular model with an adaptive radius consisting of two concentric circles. As shown in Fig. 6(b), let the geometric center p of the query cluster be the center of two concentric circles, let the maximum horizontal distance $d_{\max}$ between the geometric center p and any point inside the query cluster be the search radius of inner circle, let $d_{\max} + r$, where r is a threshold value that controls the search radius, be the search radius of outer circle. Theoretically, the number of points ($N_{\max}$) inside the outer circle is equal to the number of points ($N_{\min}$) inside the inner circle if the query cluster is part of a pole-like object. Considering a possible scenario in which the points of some non-pole-like objects such as boards of street signs might exist, we label the query cluster as a pole-like object if $N_{\max} - N_{\min} < N_{\text{threshold}}$ is satisfied (where $N_{\text{threshold}}$ is a threshold value that controls the number of non-pole-like object points).

Fig. 7 illustrates real examples of the circular model for detecting voxels of different ground objects. Utility poles are primarily composed of linear voxels without nonlinear voxels in the vertical direction and there is an almost empty space around them in the horizontal direction, whereas street tree canopies

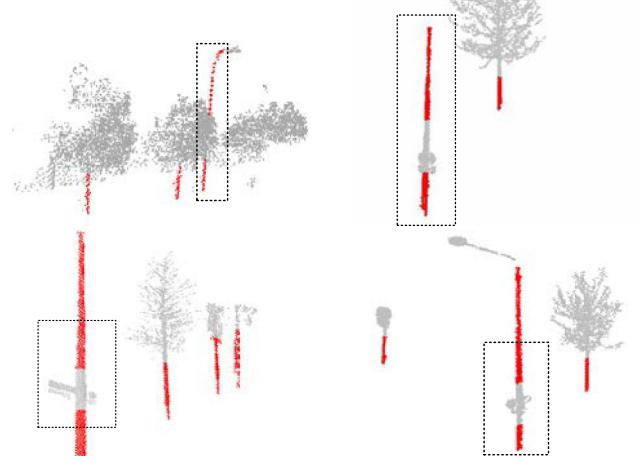


Fig. 8. Some typical linear voxels belonging to the pole-like objects. Linear voxels belonging to the pole-like objects: red, Other voxels: gray.

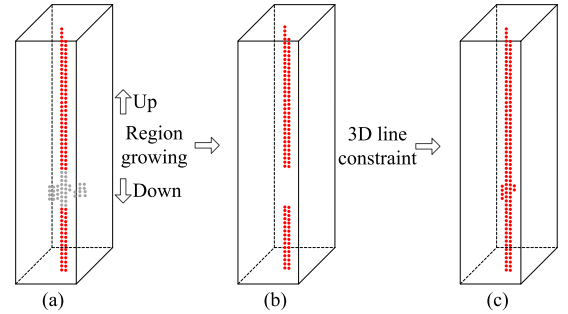


Fig. 9. Schematic diagram of extracting each individual pole-like object. Linear voxels belonging to the pole-like objects: red, Other voxels: gray.

and buildings present relatively complex structures and mostly have planar or spherical voxels in the vertical and horizontal directions.

C. Pole-Like Object Extraction Based on Vertical Region Growing

For each slice, voxels belonging to the pole-like objects are identified, which will be used as seed voxels for detecting the individual pole-like objects. As shown in Fig. 8, some components within a pole-like object might not be pole-like. To obtain the individual pole-like objects, in this section, a vertical region growing algorithm [10] is conducted within a pillar structure for matching and merging the pole-like voxels, that is, the same pole-like objects. As a result, all the individual pole-like objects are extracted. As shown in Fig. 9, the specific process of conducting the vertical region growing algorithm is as follows.

- 1) The vertical growing starts from one of the voxels belonging to pole-like objects to create the first individual pole-like object.
- 2) Vertically growing from seed voxels inside a pillar structure, the nearest voxels belonging to pole-like objects are added to the query segmented object.
- 3) The growing continues until the distance between the query segmented object and the next nearest voxel belonging to pole-like objects exceeds a threshold of 0.5 m.

This threshold is determined by the minimum distance between any two pole-like objects in the experimental scene.

- 4) The previous steps are repeated until all the voxels belonging to pole-like objects are traversed. As a result, each group of points denotes an individual pole-like object.

Due to the non-pole-like components within the pole-like objects, some of the individual pole-like objects extracted by only using the vertical region growing algorithm might not be vertically continuous [see Fig. 9(b)]. For these broken pole-like objects, we use a random sample consensus (RANSAC) algorithm [31] to fit the query individual pole-like object into a 3-D line, since it uses initial data that satisfy as few fitting conditions as possible and then adopts a consensus to expand the dataset, which is robust to noises. Then, the points at the broken parts whose distances to the 3-D fit line are less than the maximum radius are also added to the query individual pole-like object for obtaining the complete pole-like object [see Fig. 9(c)].

D. Pole-Like Object Classification Based on Semantic Rules

Road environment might consist of various types of pole-like objects, such as lamp posts, street signs, traffic lights, utility poles, and tree trunks. These different types of pole-like objects usually exhibit different morphological characteristics and spatial topological relationships. Therefore, after identifying all the individual pole-like objects generated in Section II-C, several semantic rules can be derived in terms of the morphological characteristics of an object and its spatial relationship with others and used to classify these extracted pole-like objects into four categories, i.e., tree trunks, utility poles, lamp posts, and others (e.g., street signs).

Height [32] is an important characteristic of pole-like objects and usually varies from one type to another, which can be used as criteria to classify different pole-like objects. For example, the heights of pole-like objects such as lamp posts, street signs, traffic lights, and utility poles, normally have explicit specifications, whereas those of tree trunks vary according to tree ages and species. Utility poles should be the highest objects to ensure the safety of power line and the reliable distribution of electricity.

As shown in Fig. 10, different pole-like objects exhibit different two-dimensional (2-D) projected point distributions and the red 2-D grids are positions of the individual pole-like objects extracted Section II-C. For example, tree trunks should connect to tree crowns, which can be used for identifying tree trunks. More specifically, we accumulate the number of 2-D grids containing points within 24 neighbors of the query red grids. If the number of 2-D grids containing points exceeds the threshold value, the query pole-like object is labeled as tree trunks. As a result, tree trunks are distinguished from other type of pole-like objects. For the subsequent classification procedure, utility poles, lamp posts, and others are classified based on their heights. Generally, the height ranges of utility poles and lamp posts in a specific area can be retrieved from municipal departments. In our implementation, the normalized height of each pole-like object (N_h) is calculated. If

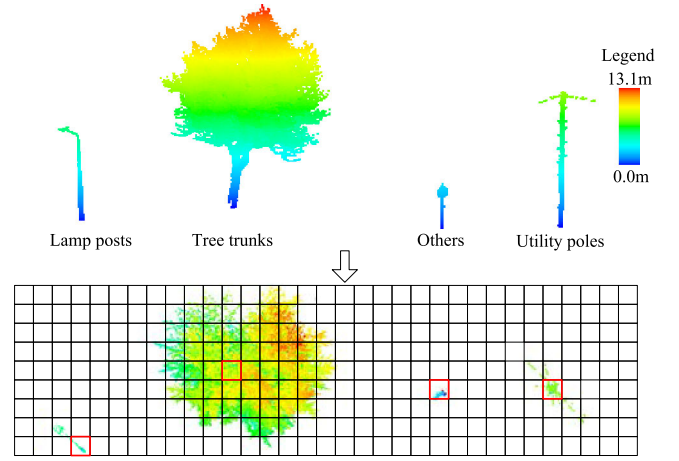


Fig. 10. 2-D projected point distributions among different pole-like objects. Point cloud data are rendered from blue to red as the height value increases. The red 2-D grids are positions of the extracted pole-like objects.

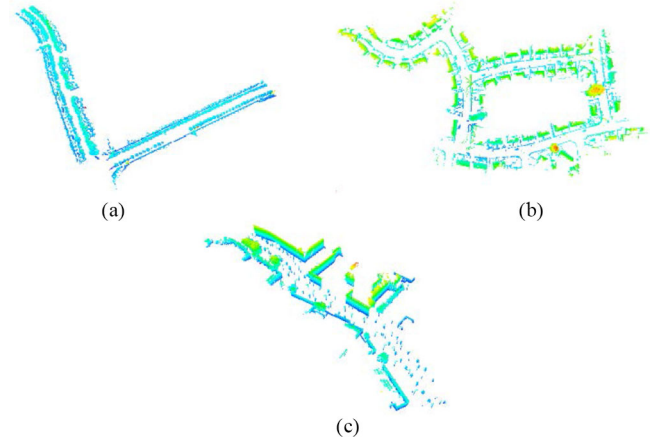


Fig. 11. Three experimental datasets of mobile LiDAR point clouds. Points are colored from blue to red as the height value increases. (a) City block (dataset I). (b) Residential area (dataset II). (c) Street scene (dataset III).

$N_h > h_{\text{utility}}$, the query pole-like object is labeled as utility poles. If $N_h > h_{\text{lamp}}$, the query pole-like object is labeled as lamp posts. If neither of these two conditions is satisfied, the query pole-like object is labeled as others.

III. EXPERIMENTATION AND ANALYSIS

To verify the effectiveness and robustness of the proposed method, we performed both qualitative and quantitative evaluations using three experimental datasets from mobile LiDAR point cloud data.

A. Experimental Data and Evaluation Criterion

In dataset I, a city block covered approximately $282 \times 302 \text{ m}^2$ and was scanned by the ROAMER system developed by the Finnish Geodetic Institute [see Fig. 11(a)]. The total number of points in dataset I was 2 128 736 points, with the point density of approximately 274 points/ m^2 . Dataset I contained large number of structures i.e., street lights, utility poles, street trees, buildings, and vehicles. In dataset II, a residential area covered

approximately $447 \times 365 \text{ m}^2$ and was scanned by the Lynx Mobile Mapper system developed by Optech [see Fig. 11(b)]. The total number of points in dataset II was 8 068 188, with the point density of approximately 1063 points/ m^2 . Dataset II comprised buildings, lamp posts, utility poles, and a very small number of street trees. In dataset III, a street scene covering approximately $430 \times 165 \text{ m}^2$ was provided by Delft University of Technology [see Fig. 11(c)]. The total number of points in dataset III was 17 466 607, with the point density of approximately 756 points/ m^2 . Dataset III mainly consisted of buildings, lamp posts, street trees, vehicles, and street signs.

To quantitatively evaluate the performance of the extraction of tree trunks, lamp posts, and utility poles, the experimental data were marked manually and used as the ground truth data, to which the results of the proposed method were compared. Then, the efficiency for extracting 3-D information about tree trunks, lamp posts, utility poles, and others was analyzed and evaluated based on the confusion matrix. This paper uses the recall (Re) and precision (Pr) as the detection evaluation indexes. The recall rate represents a measure of completeness or quantity and the precision rate represents a measure of exactness or quality. Let us assume that the original samples include positive samples and negative samples. Then, we have

$$\text{Re} = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\% \quad (9)$$

$$\text{Pr} = \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100\% \quad (10)$$

where TP denotes the number of pole-like objects that are correctly extracted as pole-like objects, FN represents the number of pole-like objects that are incorrectly extracted as others, and FP denotes the number of others that are incorrectly extracted as pole-like objects.

B. Effect of Point Density Change on the Recognition Results

For the reflectors with different distances from the emitter, the number of points each voxel (with equal size) contained might vary with its distance from the emitter. The further the sensor from the reflectors, the sparser the scanned point cloud (i.e., the smaller the number of points each voxel contains). Although the point cloud becomes sparse, the local geometric characteristic inside each voxel is constant. For reflectors with the same distance from the emitter, the resolution of point clouds is almost consistent. To investigate the effect of point density on the shape detection, we performed a stimulated experiment. In our implementation, a set of cylinder-shaped point clouds with different sampling intervals were derived to calculate the linear, planar, and spherical characteristics. Fig. 12(a)–(c) shows part of stimulated data. The relationship between the sampling interval and linear (planar or spherical) characteristics is presented in Fig. 12(e). Stimulated experiment results suggested that the change of point density barely affected the detection of the local linear structure.

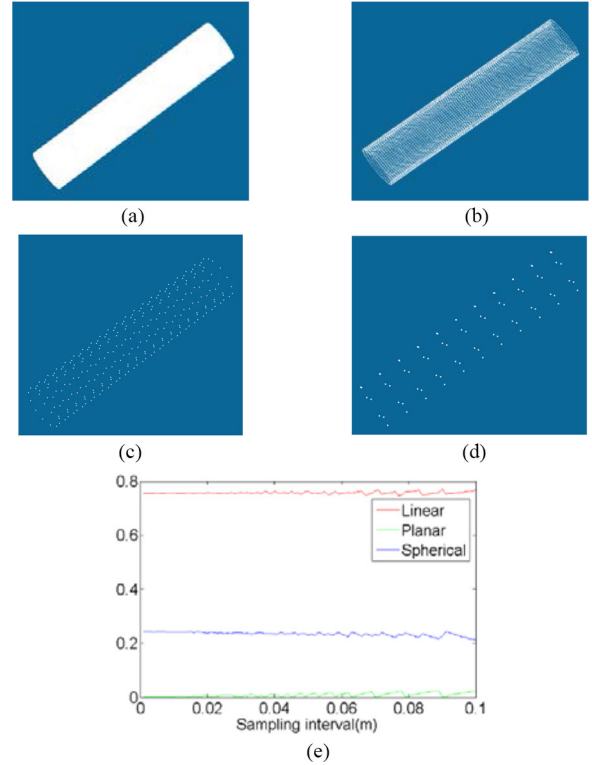


Fig. 12. Effect of point density change on the recognition results. (a) 0.001 m. (b) 0.01 m. (c) 0.05 m. (d) 0.1 m. (e) Relationship between sampling interval and local geometric characteristics.

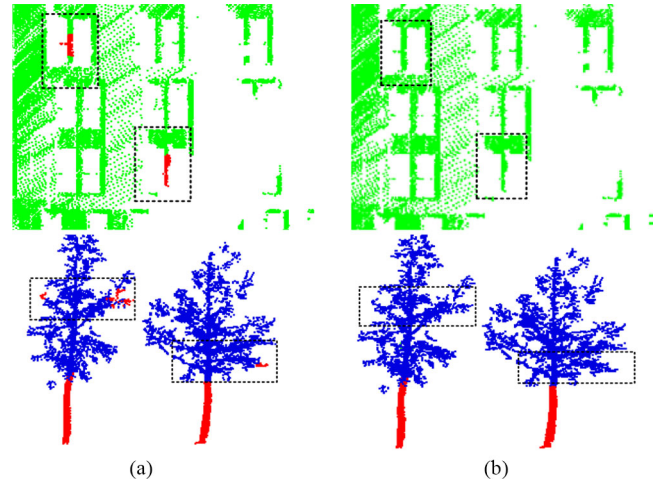


Fig. 13. Shape recognition optimization comparison using contextual information. Linear: red, Planar: green, and Spherical: blue. (a) Shape recognition without context. (b) Shape recognition with context.

C. Effect of the MRF Model on Voxel-Based Shape Recognition

As described earlier, the performance of voxel-based shape recognition directly affected the subsequent detection and classification of pole-like objects. Thus, we used the MRF model to optimize the results of PCA since some pole-like parts might belong to other ground objects, especially buildings and tree branches [see Fig. 13(a)]. Through contextual information

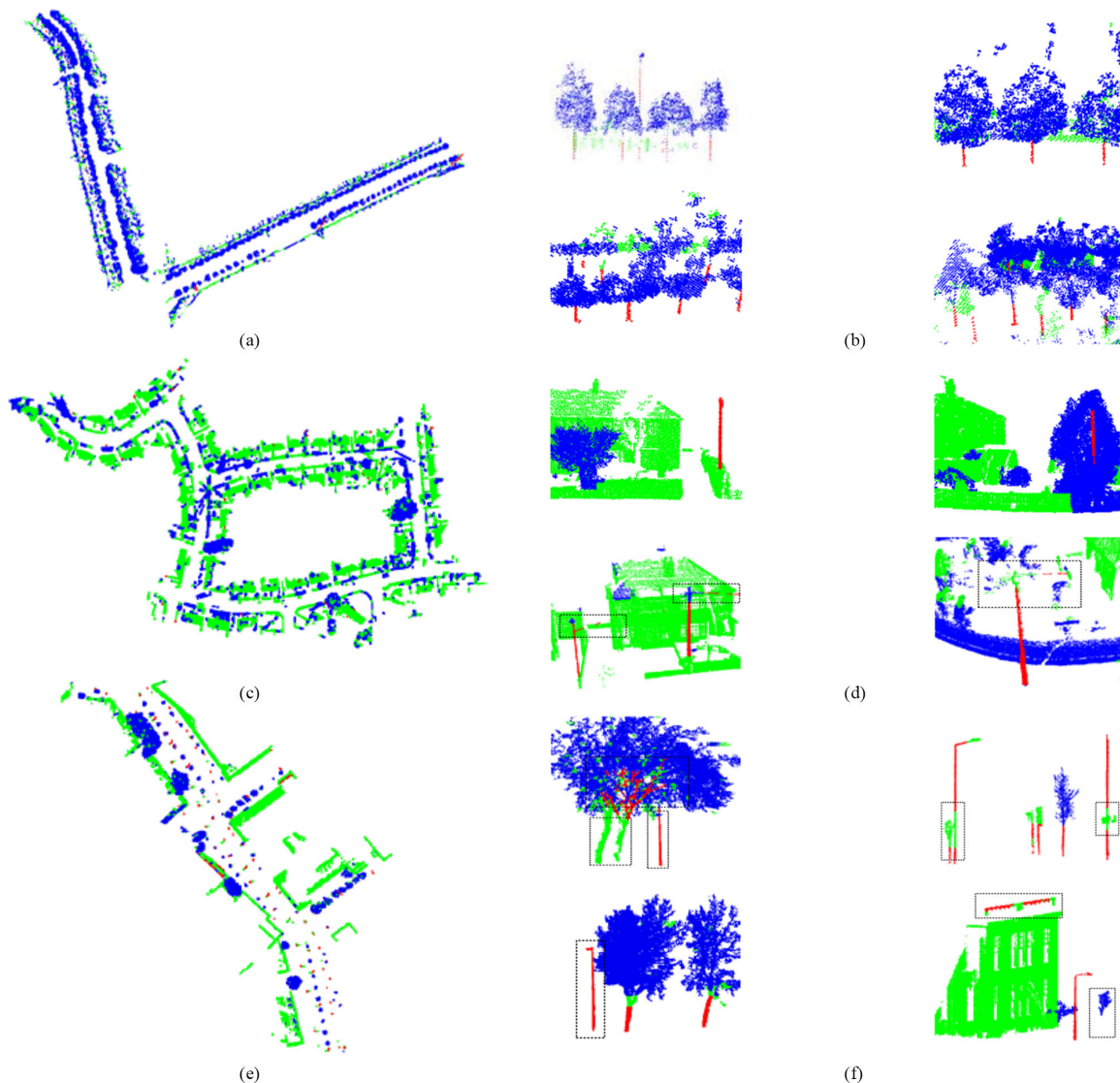


Fig. 14. Voxel-based shape recognition results of three experimental data after optimization. Linear: red, Planar: green, and Spherical: blue. (a) Overview of voxel-based shape recognition (dataset I). (b) Details of typical parts (dataset I). (c) Overview of voxel-based shape recognition (dataset II). (d) Details of typical parts (dataset II). (e) Overview of voxel-based shape recognition (dataset III). (f) Details of typical parts (dataset III).

constraint, the isolated linear parts could be smoothed if they were surrounded by planar and spherical parts. To evaluate the advantage of the MRF model, we compared the shape recognition results between context and noncontext [see Fig. 13(a) and (b)]. Experimental results indicated that pole-like parts belonging to other ground objects could be removed from the candidates, which would efficiently reduce the error rate at the subsequent extraction procedure. Fig. 14 demonstrates the voxel-based shape recognition results of the three experimental datasets after optimization, which suggested that the linear structures in scenes could be correctly inferred.

D. Pole-Like Objects Extraction Results

As shown in Fig. 14, it is also noticed that the identified linear voxels would not always be the parts of pole-like objects, such as power lines, tree branches, and linear parts of buildings. Moreover, as mentioned earlier, some components within a pole-like

object might not be pole-like, which might result in the vertical noncontinuity [see Fig. 14(f)]. Thus, the circle model was used to recognize the linear voxels belonging to the pole-like objects and the vertical region growing algorithm was conducted to individualize the pole-like objects for matching and merging the same pole-like objects. To verify the effectiveness of the proposed method for extracting the individual pole-like objects, we used the three experimental datasets as examples. In our experiments, the size of the voxels was set to 0.6 m. Fig. 15 illustrates the overview of the extracted pole-like objects from datasets I–III. When a lamp post cuts through a tree canopy, some of the lamp post points coincide with those of the tree trunk. In this situation, the proposed detection method would extract parts of the pole-like object above and below the tree canopy and then obtain the individual pole-like objects by upward and downward growing [see Fig. 15(b)]. Furthermore, parts of buildings and tree branches were eliminated from the extraction results [see Fig. 15(f)] because they did not satisfy the

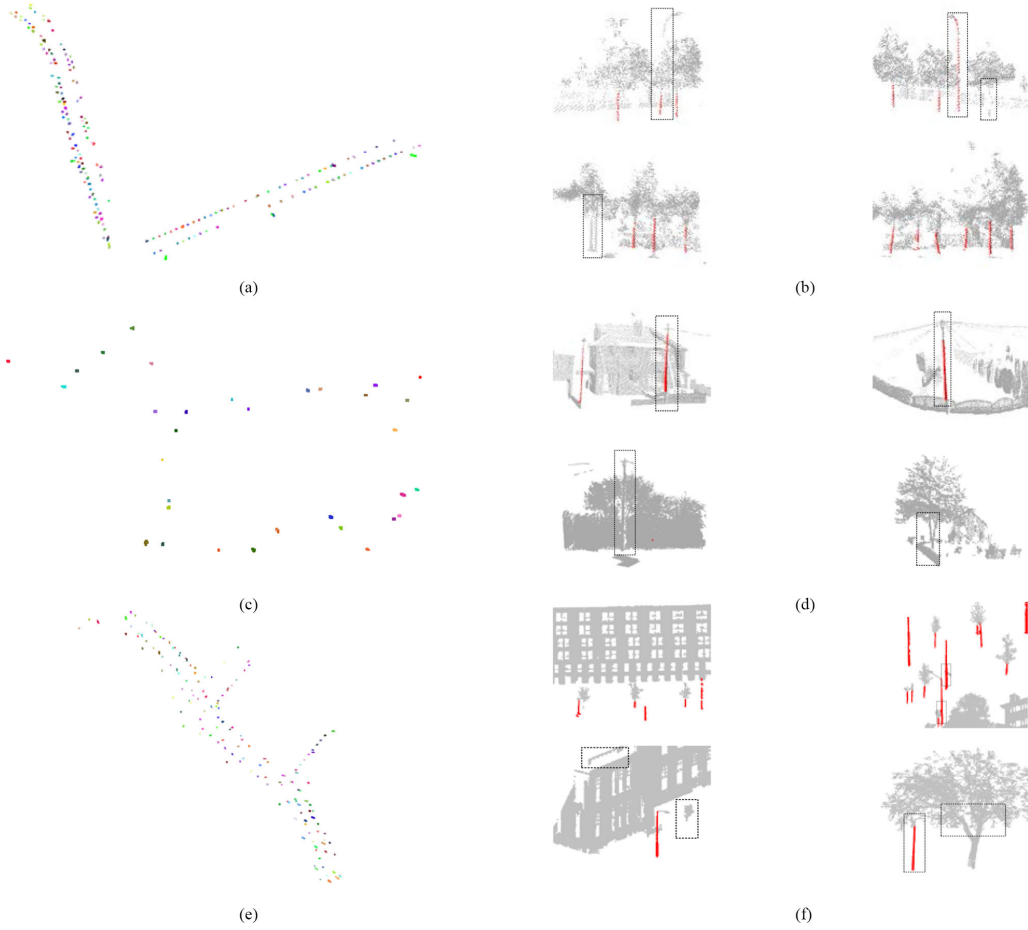


Fig. 15. Results of the extracted pole-like objects from three experimental data. The individual pole-like objects in (a), (c), and (d) are randomly colored. The pole-like objects in (b), (d), and (f) are colored in red, whereas the non-pole-like objects are rendered in gray. (a) Overview of the individual pole-like objects (dataset I). (b) Details of typical examples (dataset I). (c) Overview of the individual pole-like objects (dataset II). (d) Details of typical examples (dataset II). (e) Overview of the individual pole-like objects (dataset III). (f) Details of typical examples (dataset III).

circular model. As shown in Fig. 15(f), the broken pole-like objects were reconstructed by the RANSAC-based method to obtain whole individual pole-like objects. Nevertheless, there were several limitations to our proposed extraction method. Some of the pole-like objects failed to be extracted for the following reasons. First, the point cloud data of the tree trunk were missing [see Fig. 15(b), (d), and (f)] due to occlusions. Second, PCA might not locate the pole-like parts, such as tree trunks with large diameters [see Fig. 15(f)]. Third, the pole-like objects were close to each other, such as tree trunks [see Fig. 15(b)]. Fourth, the pole-like objects were completely surrounded by street tree crown, such as utility pole [see Fig. 15(d)]. Despite these limitations, the proposed method showed robust detection results, especially for not only detecting lamp posts through tree crown [see Fig. 15(b)] and utility poles with power lines [see Fig. 15(d)], but also removing the pole-like parts of other ground objects such as buildings [see Fig. 15(f)].

E. Pole-Like Objects Classification Results

To evaluate the effectiveness and robustness of our proposed classification method, we conducted the experiments using three datasets. Fig. 16 shows some typical classification results of

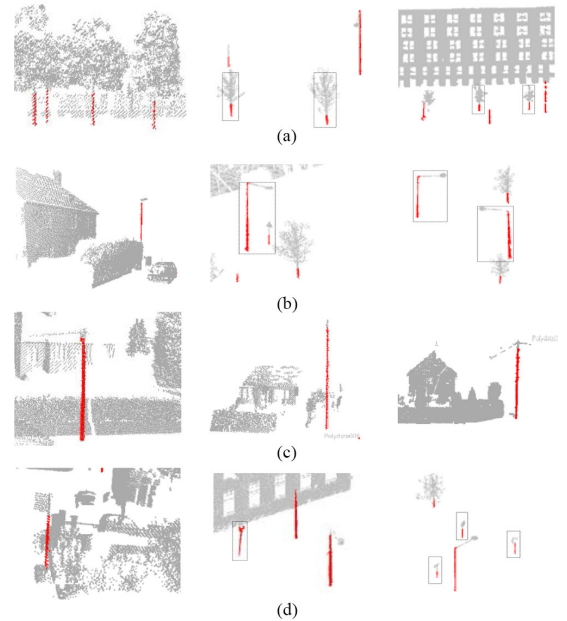


Fig. 16. Some typical classification results of the extracted pole-like objects from three datasets. The classified pole-like objects are colored in red whereas the non-pole-like objects are rendered in gray.

TABLE III
CONFUSION MATRIX OF POLE-LIKE OBJECTS CLASSIFICATION (DATASET I, EXTRACTION RATE: 85.3%)

		Manual interpretation				<i>Pr</i>
		Tree trunks	Lamp posts	Utility poles	Others	
Proposed method	Tree trunks	226	0	0	3	229 98.7%
	Lamp posts	1	6	0	0	7 85.7%
	Utility poles	0	1	3	0	4 75.0%
	Others	0	1	0	3	4
	Missing extraction	38	0	1	3	42
	Sum	265	8	4	9	286
	<i>Re</i>	85.6%	75.0%	75.0%		

TABLE IV
CONFUSION MATRIX OF POLE-LIKE OBJECTS CLASSIFICATION (DATASET II, EXTRACTION RATE: 94.1%)

		Manual interpretation				<i>Pr</i>
		Lamp posts	Utility poles	Others	Sum	
Proposed method	Lamp posts	17	1	1	19	89.5%
	Utility poles	2	9	0	11	81.8%
	Others	0	0	2	2	
	Missing extraction	0	1	1	2	
	Sum	19	11	47	34	
	<i>Re</i>	89.5%	81.8%			

TABLE V
CONFUSION MATRIX OF POLE-LIKE OBJECTS CLASSIFICATION (DATASET III, EXTRACTION RATE: 92.3%)

		Manual interpretation				<i>Pr</i>
		Tree trunks	Lamp posts	Others	Sum	
Proposed method	Tree trunks	66	0	1	67	98.5%
	Lamp posts	1	44	4	49	89.8%
	Others	0	7	34	41	
	Missing extraction	11	0	2	13	
	Sum	78	51	41	170	
	<i>Re</i>	84.6%	86.2%			

the extracted pole-like objects from the three datasets, which indicated that the proposed classification method was efficient and robust to classify most of the extracted pole-like objects. Tables III–V list the confusion matrix of the pole-like objects classification. Fig. 17 presents some typical examples of error and missing classification.

As shown in Table III, the city block dataset (dataset I) contained a total of 286 pole-like objects. Of these, the proposed method detected 226 as tree trunks, 6 as lamp posts, and 3 as utility poles. As defined in (9) and (10), the recall rates for tree trunks, lamp posts, and utility poles are 85.6%, 75.0%, and 75.0%, respectively, and the precision rates are 98.7%, 85.7%, and 75.0%, respectively. Some of the tree trunks were labeled as others due to the missing point cloud data of tree crown [see Fig. 17(a)]. A total of 244 pole-like objects were extracted from dataset I, while 42 were missing, yet an extraction rate of 85.3% was achieved for pole-like objects.

As shown in Table IV, the residential area dataset (dataset II) comprised a total of 34 pole-like objects, of which the proposed method correctly detected 17 as lamp posts (recall rate: 89.5%; precision rate: 10.5%) and 9 as utility poles (recall rate: 81.8%;

precision rate: 18.2%). Since the residential area dataset contained no structures related to street tree trunks, the proposed method extracted no street tree trunk. As shown in Fig. 17(c), the missing data of tree trunks fail to be detected and further classified. Consequently, 32 out of 34 pole-like objects were correctly labeled by the proposed method from dataset II, with a detection rate of 94.1%.

As shown in Table V, the street scene dataset (dataset III) consisted of a total of 170 pole-like objects, where the proposed method correctly extracted 66 as tree trunks (recall rate: 84.6%; precision rate: 1.5%) and 44 as lamp posts (recall rate: 86.2%; precision rate: 10.2%). In the street scene dataset, some of the lamp posts were classified as street crown since their topmost part was close to the street tree crown [see Fig. 17(b)]. In addition, the shape recognition procedure failed to locate some street trunks with large diameters, which also resulted in the missing extraction [see Fig. 17(d)]. Furthermore, the missing tree crowns might result in the wrong classification of the tree trunk. As a result, the proposed method correctly extracted 157 out of 170 pole-like objects from dataset III, with a detection rate of 92.3%.

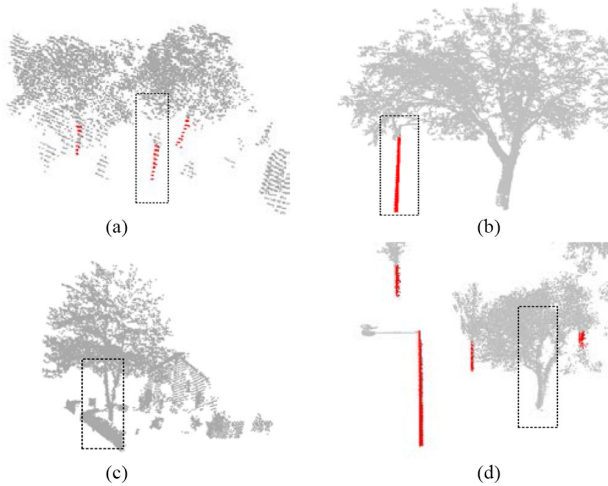


Fig. 17. Some typical examples of error and missing classification. The classified pole-like objects are colored in red, whereas the non-pole-like objects are rendered in gray. (a) Error classification #1. (b) Error classification #2. (c) Missing classification #1. (d) Missing classification #2.

IV. CONCLUSION

This paper developed a voxel-based method for automatic detection and classification of pole-like objects from mobile LiDAR point cloud data. PCA and MRF-based optimization were first used for the voxel-based dimension analysis to generate a set of linear voxels. The voxel-based shape recognition could efficiently address the problem of detecting a pole-like object in an overlapped region, from which most existing detection methods suffered. Second, the proposed circular model and the vertical region growing approach were conducted to extract and individualize the pole-like objects, which could also remove the linear voxels that did not belong to the pole-like objects. Finally, both the shape features and the spatial topological relationships were used to further classify these extracted pole-like objects. The proposed method achieved a detection accuracy of 85.3%, 94.1%, and 92.3% in the extraction of pole-like objects from three different experimental datasets.

However, voxel-based shape recognition as the first step directly affects the performance of pole-like object extraction and classification at the subsequent steps, since it might fail to locate some of the pole-like objects with large diameters, such as old tree trunks. How to be robust for finding these pole-like objects with large diameters will be the focus of our future work. Moreover, we attempt to further classify the extracted pole-like objects based on several semantic rules. Future research works will focus on introducing more information for assisting classification by integrating multisource data and exploiting machine learning algorithms.

REFERENCES

- [1] J. Huang and S. You, "Pole-like object detection and classification from urban point clouds," in *Proc. 2015 IEEE Int. Conf. Robot. Automat.*, 2015, pp. 3032–3038.
- [2] P. Doubek, M. Perdoch, and J. Matas, "Mobile mapping of vertical traffic infrastructure," in *Proc. 13th Comput. Vis. Winter Workshop*, 2008, pp. 115–122.
- [3] H. Guan, W. Yan, Y. Yu, L. Zhong, and D. Li, "Robust traffic-sign detection and classification using mobile lidar data with digital images," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 11, no. 5, pp. 1715–1724, May 2018.
- [4] Y. Yu, J. Li, H. Guan, C. Wang, and J. Yu, "Semiautomated extraction of street light poles from mobile lidar point-clouds," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 3, pp. 1374–1386, Mar. 2015.
- [5] L. Yan, Z. Li, H. Liu, J. X. Tan, S. N. Zhao, and C. J. Chen, "Detection and classification of pole-like road objects from mobile LiDAR data in motorway environment," *Opt. Laser Technol.*, vol. 97, pp. 272–283, 2017.
- [6] D. Zai, Y. Chen, J. Li, Y. Yu, C. Wang, and H. Nie, "Inventory of 3D street lighting poles using mobile laser scanning point clouds," in *Proc. Int. Geosci. Remote Sens. Symp.*, 2015, pp. 573–576.
- [7] D. A. Luo and Y. M. Wang, "Rapid extracting pillars by slicing point clouds," in *Proc. XXI ISPRS Congr.*, 2010, pp. 215–218.
- [8] S. Liberge, B. Soheilian, N. Chehata, and N. Paparoditis, "Extraction of vertical objects in 3D point clouds," in *Proc. ISPRS Tech. Commission III Symp. Photogramm. Comput. Vis. Image Anal.*, 2010, pp. 126–130.
- [9] Y. Hu, X. Li, J. Xie, and L. Guo, "A novel approach to extracting street lamps from vehicle-borne laser data," in *Proc. IEEE Int. Conf. Geoinform.*, 2011, pp. 1–6.
- [10] S. I. El-Halawany and D. D. Lichti, "Detecting road poles from mobile terrestrial laser scanning data," *Mapping Sci. Remote Sens.*, vol. 50, no. 6, pp. 704–722, 2013.
- [11] H. Zheng, F. Tan, and R. Wang, "Pole-like object extraction from mobile lidar data," *ISPRS, Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. XLI-B1, pp. 729–734, 2016.
- [12] L. Matti, A. Jaakkola, H. Juha, A. Kukko, and H. Kaartinen, "Detection of vertical pole-like objects in a road environment using vehicle-based laser scanning data," *Remote Sens.*, vol. 2, no. 3, pp. 641–664, 2010.
- [13] M. Lehtomäki, A. Jaakkola, J. Hyypä, A. Kukko, and H. Kaartinen, "Performance analysis of a pole and tree trunk detection method for mobile laser scanning data," in *Proc. ISPRS Workshop Laser Scanning*, 2012, pp. 197–202.
- [14] C. Cabo, C. Ordoñez, S. García-Cortés, and J. Martínez, "An algorithm for automatic detection of pole-like street furniture objects from mobile laser scanner point clouds," *ISPRS J. Photogramm.*, vol. 87, no. 1, pp. 47–56, 2014.
- [15] T. A. Teo and C. M. Chiu, "Pole-like road object detection from mobile lidar system using a coarse-to-fine approach," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 8, no. 10, pp. 4805–4818, Oct. 2015.
- [16] J. Wang, R. Lindenbergh, and M. Menenti, "SigVox - A 3D feature matching algorithm for automatic street object recognition in mobile laser scanning point clouds," *ISPRS J. Photogramm.*, vol. 128, pp. 111–129, 2017.
- [17] B. Rodríguez-Cuenca, S. García-Cortés, C. Ordóñez, and M. C. Alonso, "Automatic detection and classification of pole-like objects in urban point cloud data using an anomaly detection algorithm," *Remote Sens.*, vol. 7, no. 10, pp. 12680–12703, 2015.
- [18] F. Li, S. O. Elberink, and G. Vosselman, "Semantic labelling of road furniture in mobile laser scanning data," *ISPRS, Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. XLII-2/W7, pp. 247–254, 2017.
- [19] H. Yokoyama, H. Date, S. Kanai, and H. Takeda, "Detection and classification of pole-like objects from mobile laser scanning data of urban environments," *Int. J. CAD/CAM*, vol. 13, no. 2, pp. 31–40, 2013.
- [20] K. Fukano and H. Masuda, "Detection and classification of pole-like objects from mobile mapping data," *ISPRS Ann. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 2, pp. 57–64, 2015.
- [21] B. Riveiro, L. Díaz-Vilariño, B. Conde-Carnero, M. Soilán, and P. Arias, "Automatic segmentation and shape-based classification of retro-reflective traffic signs from mobile lidar data," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 9, no. 1, pp. 295–303, Jan. 2017.
- [22] F. Li, S. Oude Elberink, and G. Vosselman, "Pole-like road furniture detection and decomposition in mobile laser scanning data based on spatial relations," *Remote Sens.*, vol. 10, no. 4, 2018, Art. no. 531.
- [23] L. Li, Y. Li, and D. Li, "A method based on an adaptive radius cylinder model for detecting pole-like objects in mobile laser scanning data," *Remote Sens. Lett.*, vol. 7, no. 3, pp. 249–258, 2016.
- [24] J. Demantké, C. Mallet, N. David, and B. Vallet, "Dimensionality based scale selection in 3D LIDAR point clouds," *ISPRS, Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 3812, no. 5, pp. 97–102, 2012.
- [25] B. Yang and Z. Dong, "A shape-based segmentation method for mobile laser scanning point clouds," *ISPRS J. Photogramm.*, vol. 81, no. 7, pp. 19–30, 2013.

- [26] Y. Wang, L. Cheng, Y. Chen, Y. Wu, and M. Li, "Building point detection from vehicle-borne lidar data based on voxel group and horizontal hollow analysis," *Remote Sens.*, vol. 8, no. 5, 2016, Art. no. 419.
- [27] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-6, no. 6, pp. 721–741, Nov. 1984.
- [28] Z. Li *et al.*, "A three-step approach for TLS point cloud classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 9, pp. 5412–5424, Sep. 2016.
- [29] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, 2002, vol. 1, pp. 377–384.
- [30] U. V. Luxburg, S. Bubeck, S. Jegelka, and M. Kaufmann, "Nearest neighbor clustering," *Ann. Statist.*, vol. 9, no. 1, pp. 135–140, 1981.
- [31] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [32] A. Brunn and U. Weidner, "Extracting buildings from digital surface models," *Int. Arch. Photogramm. Remote Sens.*, vol. 32, no. 3, pp. 27–34, 1997.



Zhizhong Kang received the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2004.

He was a Postdoctoral Researcher with Delft University of Technology, Delft, The Netherlands, from 2006 to 2008. He is currently a Full Professor and the Head with the Department of Remote Sensing and Geo-Information Engineering, China University of Geosciences, Beijing, China. He has authored more than 60 refereed journal and conference publications.

His research interests include digital photogrammetry, LiDAR data processing, indoor modeling and navigation, and planetary remote sensing.

Dr. Kang has served as the Chair of ISPRS WG IV/5: Indoor/Outdoor seamless modelling, LBS, mobility. He was a recipient of the 2015 ERDAS Award for Best Scientific Paper in Remote Sensing by American Society for Photogrammetry and Remote Sensing.



Juntao Yang received the Master's degree in LiDAR technology in 2017 from the China University of Geosciences, Beijing, China, where he currently working toward the Ph.D. degree at the School of Land Science and Technology.

His research interests include LiDAR data processing, computer vision, and lunar structure recognition and analysis.



Ruofei Zhong received the Ph.D. degree in geoinformatics from the Chinese Academy of Science's Institute of Remote Sensing Applicants, Beijing, China, in 2005.

He is currently a Professor with Beijing Advanced Innovation Center for Imaging Technology and Key Lab of 3D Information Acquisition and Application, College of Resource Environment and Tourism, Capital Normal University, Beijing, China. His research interests include LiDAR data processing and data collection system with laser scanning.



Yongxing Wu received the Master's degree in LiDAR technology from the School of Land Science and Technology, China University of Geosciences, Beijing, China, in 2018.

Currently, he is working with Shenzhen Investigation & Research Institute Co., Ltd., Shenzhen, China. His research interest focuses on LiDAR data processing.



Zhenwei Shi received Bachelor's and Master's degrees from the China University of Geosciences, Beijing, China, in 2013 and 2016, respectively. He is currently working toward the Ph.D. degree in photogrammetry and remote sensing at Peking University, Beijing, China.

His research interests include LiDAR data processing, object recognition from point clouds, and LiDAR remote sensing.



Roderik Lindenbergh received the Ph.D. degree in topology and geometry from the Mathematical Institute, Utrecht University, Utrecht, The Netherlands, in 2002.

Since then he is working for what is currently called the Department of Geoscience and Remote Sensing, Delft University of Technology, Delft, The Netherlands. He is an Assistant Professor and his main research interests include laser scanning and digital terrain analysis.

Dr. Lindenbergh is the 2016–2020 Chair of ISPRS WG II/10 on environmental and structural monitoring and was the Chair of the Laser Scanning Workshop at the ISPRS 2017 Geospatial Week in Wuhan, China.