Bag of Contextual-Visual Words for Road Scene Object Detection From Mobile Laser Scanning Data

Yongtao Yu, *Member, IEEE*, Jonathan Li, *Senior Member, IEEE*, Haiyan Guan, *Member, IEEE*, Cheng Wang, *Senior Member, IEEE*, and Chenglu Wen, *Member, IEEE*

***Abstract*—This paper proposes a novel algorithm for detecting road scene objects (e.g., light poles, traffic signposts, and cars) from 3-D mobile-laser-scanning point cloud data for transportation-related applications. To describe local abstract features of point cloud objects, a contextual visual vocabulary is generated by integrating spatial contextual information of feature regions. Objects of interest are detected based on the similarity measures of the bag of contextual-visual words between the query object and the segmented semantic objects. Quantitative evaluations on two selected data sets show that the proposed algorithm achieves an average recall, precision, quality, and F-score of 0.949, 0.970, 0.922, and 0.959, respectively, in detecting light poles, traffic signposts, and cars. Comparative studies demonstrate the superior performance of the proposed algorithm over other existing methods.**

***Index Terms*—Bag-of-contextual-visual-words, car, light pole, mobile laser scanning (MLS), road scene object, traffic signpost.**

I. Introduction

With rapid urbanization, effective management and maintenance of urban road facilities (e.g., light poles, traffic signposts, etc.), on a regular basis, play a critical role in providing convenient and safe driving environments to the road users. To facilitate management and improve efficiency, automated, cost-effective detection and measurement of road scene objects are urgently demanded by the transportation agencies. Accurate category and localization information of road scene objects also forms important inputs to many intelligent transportation-related applications, including driver assistance and safety warning systems [1], [2], autonomous driving [3]–[5], and traffic flow monitoring and prediction [6]–[8]. In addition, accurate, real-time information regarding current road conditions, traffic flow, and the surrounding environment is of great significance and necessity to the Intelligent Transportation Systems.

Traditionally, the statistics, localization, and measurement of road scene objects were mainly accomplished based on field work, where field specialists from transportation agencies conducted on-site inspections and measurements on a regular basis. Such field measurements were time consuming, labor intensive, costly, and inefficient to inventory large-scale, complicated urban road networks. Specifically, it is even greatly dangerous to work on highways, overhead roads, or in tunnels. Recently, mobile mapping systems (MMS) mounted with digital camera(s) or video camera(s) [9], [10] have emerged as an effective, promising tool for a wide range of transportation-related activities. However, optical imaging-based MMS suffer greatly from environmental illumination conditions, thereby limiting mapping missions to only the daytime. In addition, distortions, motion blurs, and color imbalance of images, occlusions caused by nearby pedestrians and moving vehicles, shadows cast by buildings and trees, and lack of accurate georeferencing also limit the use of the MMS in transportation-related activities.

Since last decade, with the integration of laser scanning and position and orientation technologies, mobile laser scanning (MLS) systems have emerged and been widely used in fields of transportation, road planning, heritage documentation, forestry, and basic surveying and mapping. MLS systems [11] adopt an active sensing pattern to measure surface topologies of visual targets with near-infrared laser spectra. Compared with optical imaging-based MMS, MLS systems are immune to the impact of environmental illumination conditions, and can acquire highly dense and accurate undistorted 3-D point cloud data with real-world coordinates and reflected intensities over a large area in a short time period. Therefore, MLS systems provide a potential and promising solution to detecting road scene objects to assist in rapid update of road databases and intelligent transportation-related applications. However, there are still some challenges in handling 3-D point clouds toward effective road scene object detection, such as irregular sampling, object variations, incompleteness of objects caused by occlusions, density variations caused by different sensor-object distances, non-equal noise/error levels, distortions of moving objects, etc.

In this paper, we propose a bag-of-contextual-visual-words (BoCVWs) model for detecting road scene objects from MLS

Manuscript received July 17, 2015; revised January 29, 2016; accepted March 30, 2016. This work was supported in part by the National Natural Science Foundation of China under Grant 41471379, by the Priority Academic Program Development (PAPD), and by the Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET). The Associate Editor for this paper was H. Huang. *(Corresponding author: Jonathan Li.)*

Y. Yu is with Fujian Key Laboratory of Sensing and Computing for Smart Cities, Xiamen University, Xiamen 361005, China, and also with the Faculty of Computer and Software Engineering, Huaiyin Institute of Technology, Huaian 223003, China (e-mail: allennessy.yu@gmail.com).

J. Li is with Fujian Key Laboratory of Sensing and Computing for Smart Cities, Xiamen University, Xiamen 361005, China, and also with the Department of Geography and Environmental Management, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: junli@xmu.edu.cn; junli@uwaterloo.ca).

H. Guan is with the College of Geography and Remote Sensing, Nanjing University of Information Science and Technology, Nanjing 210044, China (e-mail: guanhy.nj@nuist.edu.cn).

C. Wang and C. Wen are with Fujian Key Laboratory of Sensing and Computing for Smart Cities, Xiamen University, Xiamen 361005, China (e-mail: cwang@xmu.edu.cn; clwen@xmu.edu.cn).

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/TITS.2016.2550798

data. The proposed algorithm includes two stages: contextual visual vocabulary generation and road scene object detection. At the contextual visual vocabulary generation stage, first, the training data are preprocessed to remove ground points. Then, the off-ground points are supervoxelized to construct feature regions based on first-order supervoxel neighbors. Next, contextual feature groups are formulated and a distance metric is designed to measure their distances. Finally, the contextual feature groups are clustered and quantized to generate a contextual visual vocabulary, where each cluster center forms a distinct contextual visual word. At the road scene object detection stage, first, a search scene is preprocessed to filter out ground points. Then, the off-ground points are clustered and segmented into individual semantic objects through Euclidean distance clustering and extended voxel-based normalized cut segmentation. Next, the query object and the semantic objects are supervoxelized, featured, and quantized to form BoCVWs. Finally, the objects of interest are detected based on the similarity measures between the BoCVWs of the query object and the semantic objects.

The contributions of this paper are as follows: 1) a supervoxel-based BoCVWs model is proposed for representing point cloud objects; 2) an extended voxel-based normalized cut segmentation method is developed for segmenting connected and overlapped semantic objects.

II. Related Work

MLS systems have realized rapid collection of 3-D geospatial data used for a wide range of transportation-related applications [12]. The highly dense and accurate 3-D point cloud data have been a leading source for highway mapping [13], urban road distress assessment [14], [15], and road feature inventory [16], [17]. This trend is keeping increasing. In the following sections, we present a detailed literature review of existing methods for detecting light poles, traffic signposts, cars, and other road scene objects from MLS data.

A. Light Pole Detection

Light poles, a typical kind of road infrastructure, are an important component of the city lighting system. On one hand, light poles provide illumination to pedestrians and vehicles at night for a clear visibility of the road environment. On the other hand, light poles can effectively reduce criminal activities and terrible accidents at night for a safe driving and living environment. Most of existing methods for detecting light poles basically consider their pole-like structures.

By considering both shape and context features of light poles, a point classification based on principal component analysis (PCA) [18] was proposed to detect light poles. Similarly, an eigenvalue analysis-based method was proposed in [19] to detect light poles. Generally, the eigenvalue-based methods show high computational efficiencies. However, caused by the interference of other pole-like objects (e.g., trees, utility poles, and traffic signposts) in the scene, such methods often generate many false alarms. By using prior knowledge of shape, height, and size of light poles, a percentile-based method was developed in [16]. In this method, considering the impact of the bottom shrubs, as well as other attachments to the pole (e.g., advertising boards and traffic signs), the third quartile was selected and sliced into horizontal profiles for recognizing pole-like structures.

In [20], a pairwise 3-D shape context descriptor was developed to detect light poles. The detection of light poles was achieved through a prototype-based shape matching. A 3-D object matching framework was proposed in [17] for detecting light poles of varying shapes, completeness, with attachments, or hidden in trees. In [21], a voxel structure-based method was developed to detect light poles. The recognition of light poles was accomplished through 3-D voxel neighborhood analysis.

Currently, to simplify data processing, some research converted 3-D point clouds into 2-D representations. In [22], a 2-D point density segmentation method was proposed to detect light poles. In [23], the points of an object were first projected onto a horizontal plane. Then, the distribution of the projected points was analyzed to detect objects with pole-like structures. A density of projected points method was used in [24] to detect light poles. In addition, scan line-based methods [25]–[27] were also developed for detecting light poles.

B. Traffic Signpost Detection

As a greatly important transportation infrastructure, traffic signposts play a critical role in transportation, traffic safety, and route guidance. First, traffic signposts provide road users with detailed road information. Second, traffic signposts function to regulate and control traffic activities. Thus, detection and measurement of traffic signposts have attracted increasing attention in the literature. Most of existing methods are based on the pole-like, vertical plane, and high retro-reflectivity features of traffic signposts.

Generally, to provide optimal views and clear visibilities to the road users, traffic signposts are placed on the sides of and near the boundaries of the road. Consequently, such prior knowledge was used in [23] to detect traffic signposts. In this method, traffic signposts were detected by inferring linear features from the horizontal projection of clustered spatial objects. A Laplacian smoothing and PCA method was developed in [28] for detecting traffic signposts. In [29], a Hough forest model with a circular voting strategy was used to detect traffic signposts. Similarly, a supervoxel neighborhood-based Hough forest framework [30] was also proposed to detect traffic signposts. In addition, a LiDAR and vision-based real-time traffic signpost detection method [31] was developed for intelligent vehicle applications.

Traffic signposts usually exhibit high retro-reflectivity in the MLS data. Consequently, intensity information becomes an important clue for distinguishing traffic signposts from other objects. In [32], a processing chain of retro-intensity filtering, elevation filtering, lateral offset filtering, point regrouping, and hit count filtering was developed to detect traffic signposts. In [33], first, a point cloud was segmented into isolated objects. Then, eigenvalue analysis was performed to extract objects with linear structures. Finally, retro-reflectivity properties of traffic signposts were considered to refine the detection results. In

[34], intensity information and vertical plane structures were utilized to detect traffic signposts. Similarly, a template-driven method was developed in [35] to detect traffic signposts with the prior knowledge of symmetric shapes and highly reflective planes perpendicular to the direction of travel.

C. Car Detection

Cars are very common and important tools in current transportation activities. Detection of cars provides essential information to a variety of applications such as traffic flow monitoring, intelligent transportation, autonomous driving, business analysis of shopping malls, etc. Existing methods for car detection are basically divided into the following three categories: 1) segmentation and feature recognition based methods, 2) model-driven methods, and 3) machine learning based methods.

In [36], a marked point process based method was proposed to detect cars in crowded urban areas. In this method, a marked point process of 2-D rectangles, simulated by a multiple birth-and-death algorithm [37], was configured to describe the positions, sizes, and orientations of cars. Similarly, two-level point processes of rectangles [38] were also developed to detect cars. In [39], on-road cars were located based on detecting the changes of slopes on the transversal profiles of the road. In [40], a context-guided method was developed to detect cars based on the geometric model of cars.

In [41], an adaptive 3-D segmentation method was proposed to detect cars for motion state and velocity estimation. The detection of cars was achieved using a binary classification based on object-oriented features. In [42], an object-based point cloud analysis method was proposed for car detection. In this method, first, 3-D connected component analysis was performed to generate potential car candidates. Then, cars were detected based on area, rectangularity, and elongatedness features. In addition, Hough forest frameworks [29], [30], 3-D object matching [43], invariant parameters of polar line-segments [44], grid-cell method [45], and bottom-up and top-down descriptors [46] were also exploited for car detection.

D. Detection of Other Road Scene Objects

In addition to light poles, traffic signposts, and cars, a number of methods for detecting other common road scene objects, such as buildings, trees, utility poles, etc., have also been exploited in the literature. In [47], a mathematical morphology and supervised learning method was proposed to detect, segment, and classify urban scene objects. In this method, the entire processing was carried out using elevation images generated from 3-D point clouds. In [48], a fully automated and versatile semantic labeling framework composed of neighborhood selection, feature extraction, feature selection, and classification was developed to segment point clouds into semantic objects. A super-segments based method was proposed in [49] to interpret an urban street scene into semantic objects. Similarly, a supervoxel segmentation based approach was used in [50] for classifying urban scene objects. In addition, some other segmentation and classification methods were also exploited to detect and classify road scene objects [51]–[53].

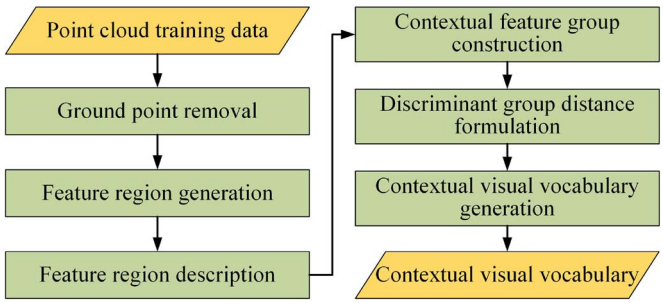


Fig. 1. Contextual visual vocabulary generation workflow.

In [54], a PCA and random sampling consensus method was proposed to detect building facades. The detected building facades were further projected onto the ground to generate building footprints. Region growing and Hough transform were used in [55] to extract vertical walls for solar energy assessment. In [56], a pairwise 3-D shape context descriptor was adopted to model the entire structures of point cloud objects. The extraction of trees was achieved based on a prototype-based shape matching process. A deep learning based method was developed in [57] for extracting and classifying urban road trees. In addition, some studies have been conducted to detect power lines [58], road markings [59], and road manhole covers [15] from MLS point clouds.

However, there are still some problems in the existing methods. On one hand, the existing methods lack of effective object-oriented descriptors to model the entire features of point cloud objects. On the other hand, the existing methods still cannot obtain promising performance when handling overlapped objects, objects of varying sizes, objects of varying geometric topologies, and objects of different levels of data incompleteness. Therefore, it is greatly important to exploit new techniques to solve the above problems toward effective point cloud object detection. In this paper, we propose a BoCVWs model for representing point cloud objects, which can effectively model the abstract features of point cloud objects. The proposed road scene object detection framework based on the BoCVWs model shows promising performance in dealing with overlapped objects, objects of varying sizes, objects of varying geometric topologies, and objects of different levels of data incompleteness.

III. Contextual Visual Vocabulary Generation

In this section, we present the technical and implementation details for the supervoxel-based contextual visual vocabulary generation from MLS point clouds (see Fig. 1). Such a contextual visual vocabulary can be further used to construct BoCVWs for representing 3-D point cloud objects.

A. Training Data Preprocessing

To generate the contextual visual vocabulary, we randomly select a group of training data, each of which has a road

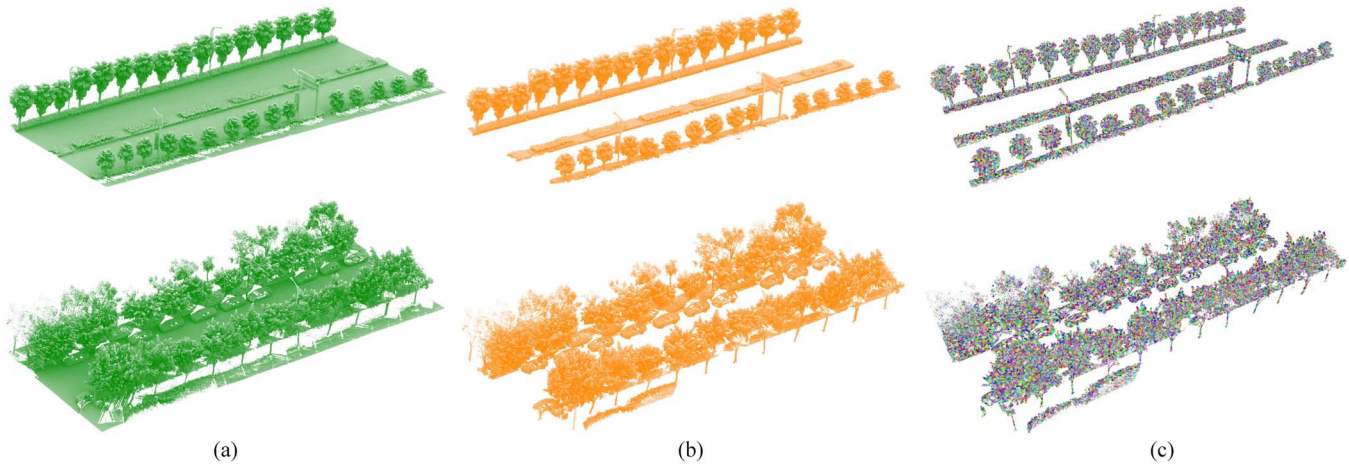


Fig. 2. Two typical training samples. (a) Raw point clouds. (b) Off-ground points obtained after voxel-based upward growing filtering. (c) Supervoxelization results of off-ground points.

segment of approximately 50 m, from the collected MLS point clouds. Fig. 2(a) shows two typical samples of the selected training data. Due to the properties of MLS systems in direct ground views and high laser measurement rates, ground points account for a great portion of the collected point clouds in a survey scene. Such large-volume ground points almost exist in all scenes and contribute very little to the generation of the contextual visual vocabulary, since the objects of interest are usually off-ground objects. Therefore, a preprocessing is first performed on the training data to remove ground points.

In our previous study, we develop a voxel-based upward growing filtering method [17] that can rapidly and effectively filter out ground points from a raw point cloud. First, considering the ground fluctuations, a point cloud is vertically divided into a set of data blocks, which are processed separately to remove ground points. Then, each of the data blocks is voxelized based on the octree partition structure. Finally, an upward growing strategy is applied to the voxels to label them into ground and non-ground voxels. The points in the ground voxels are regarded as ground points and further removed. This method has the capabilities of effectively handling large scenes with strong ground fluctuations and preserving the completeness of off-ground objects from their bottoms. Thus, in this paper, we adopt this voxel-based upward growing filtering method [17] to remove ground points from the training data. Fig. 2(b) shows the visual examples of the off-ground points obtained after voxel-based upward growing filtering.

B. Feature Region Generation

In this paper, we propose a supervoxel over-segmentation strategy to generate feature regions from the training data. Such feature regions form salient and distinctive local geometric representations of objects in the scene. To this end, each of the training data is first over-segmented into supervoxels using the voxel cloud connectivity segmentation (VCCS) algorithm [60]. There are two important parameters in the VCCS algorithm: voxel resolution and seed resolution. The voxel resolution is used to construct the voxel-cloud space, which is a simplification of the continuous point-cloud space; whereas the seed resolution is used to select seed points for constructing initial supervoxels. Fig. 2(c) shows the supervoxelization results using the VCCS algorithm with a voxel resolution of 0.05 m and a seed resolution of 0.1 m, respectively. Then, after supervoxelization, an adjacency graph $\boldsymbol{G} = \{\boldsymbol{V}, \boldsymbol{E}\}$ is constructed for all supervoxels in each of the training data, as shown in Fig. 3(a). In the adjacency graph $\boldsymbol{G}$, the vertices $\boldsymbol{V} = \{v_i\}$ are represented by the supervoxel centers; the edges $\boldsymbol{E} = \{e_{ij}\}$ are directly connected between each pair of neighboring supervoxels.

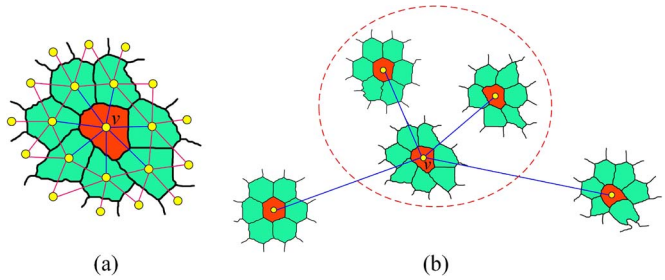


Fig. 3. (a) Adjacency graph construction and feature region generation. (b) Contextual feature group construction.

For two supervoxels centered at v_i and v_j, we define the graph distance between v_i and v_j as follows:

$$\mathrm{dis}(v_i, v_j; \boldsymbol{G}) = \begin{cases} 1, & e_{ij} \in \boldsymbol{E} \\ \min\limits_k \mathrm{dis}(v_j, v_k; \boldsymbol{G}) + 1, & e_{ij} \notin \boldsymbol{E} \wedge e_{ik} \in \boldsymbol{E}. \end{cases} \quad (1)$$

This graph distance defines the minimum number of edges connecting supervoxels v_i and v_j. Then, as shown in Fig. 3(a), based on the adjacency graph and the graph distance metric, for each supervoxel centered at v, the associated feature region is defined as a supervoxel set containing supervoxel v and its first-order neighbors on the adjacency graph. Here, the first-order neighbors of supervoxel v is defined as follows:

$$N_1(v; \boldsymbol{G}) = \{v_j | \mathrm{dis}(v, v_j; \boldsymbol{G}) = 1,\ v_j \in \boldsymbol{V}\} \quad (2)$$

that is the supervoxels directly connected to supervoxel v. The center of the feature region is assigned as the center of supervoxel v. As demonstrated in [30], such a feature region generation strategy by embedding first-order supervoxel neighbors achieves higher saliencies and distinctiveness than directly treating single supervoxels as feature regions.

C. Feature Region Description

We propose a structural-spectral descriptor to describe feature regions. This structural-spectral descriptor encodes three kinds of information for each feature region. Specifically, for a feature region centered at v, its corresponding structural-spectral descriptor is a triad $P_v = (f_v, o_v, s_v)$, where f_v represents a 20-dimensional (20-D) feature vector for modeling both geometric and intensity characteristics of feature region v; o_v denotes the orientation of feature region v; s_v is the scale of feature region v.

First, f_v contains five main components and is defined as $f_v = (a_{1D}, a_{2D}, a_{3D}, h_{\mathrm{FPFH}}, I_n)$, where a_{1D}, a_{2D}, and a_{3D} are the linear, planar, and volumetric geometric features, respectively [61]; h_{FPFH} is a 16-dimensional fast point feature histograms (FPFH) descriptor [62]; I_n is the interpolated normalized intensity. FPFH has been proven to be a promising descriptor for rapidly and saliently depicting discrete 3-D point clouds. Thus, in this paper, the FPFH descriptor is selected as a component for designing f_v. To compute a_{1D}, a_{2D}, and a_{3D}, first, we construct a covariance matrix for the points in feature region v as follows:

$$\mathbf{C}_{3\times 3} = \frac{1}{n_v}\sum_{k=1}^{n_v}(p_k - \overline{p}_c)(p_k - \overline{p}_c)^T \tag{3}$$

where n_v is the number of points in feature region v; p_k is the kth point in feature region v; and

$$\overline{p}_c = \frac{1}{n_v}\sum_{k=1}^{n_v} p_k \tag{4}$$

stands for the centroid of the points in feature region v. Then, after eigenvalue decomposition on covariance matrix $\mathbf{C}_{3\times 3}$, we obtain three eigenvalues λ_1, λ_2, and λ_3 ($\lambda_1 \geq \lambda_2 \geq \lambda_3 > 0$). Finally, the linear, planar, and volumetric geometric features are defined using these eigenvalues as follows:

$$a_{1D} = \frac{\sqrt{\lambda_1} - \sqrt{\lambda_2}}{\sqrt{\lambda_1}} \tag{5}$$

$$a_{2D} = \frac{\sqrt{\lambda_2} - \sqrt{\lambda_3}}{\sqrt{\lambda_1}} \tag{6}$$

$$a_{3D} = \frac{\sqrt{\lambda_3}}{\sqrt{\lambda_1}}. \tag{7}$$

The interpolated normalized intensity $I_n \in [0, 1]$ is computed based on the normalized intensities of the points in feature region v, and it takes the following form:

$$I_n = \frac{\sum_{k=1}^{n_v} w_k I_k}{\sum_{k=1}^{n_v} w_k} \tag{8}$$

where $I_k \in [0, 1]$ is the normalized intensity of the kth point in feature region v; w_k is the intensity weight of I_k and it is computed as follows:

$$w_k = \frac{I_k - I_{\min}}{I_{\max} - I_{\min}} \tag{9}$$

where $I_{\min}$ and $I_{\max}$ are the minimum and maximum normalized intensities in feature region v, respectively. In this way, the points with higher normalized intensities contribute more to the calculation of I_n.

Second, the orientation o_v is determined using the scatter matrix of the points in feature region v. To this end, first, we construct a scatter matrix for the points in feature region v as follows:

$$\mathbf{S}_{3\times 3} = \frac{1}{n_v}\sum_{k=1}^{n_v}(p_k - c_v)(p_k - c_v)^T \tag{10}$$

where c_v denotes the center of feature region v. Then, through eigenvalue decomposition on $\mathbf{S}_{3\times 3}$, we obtain three eigenvalues λ_1, λ_2, and λ_3 ($\lambda_1 \geq \lambda_2 \geq \lambda_3$) and the associated eigenvectors e_1, e_2, and e_3. Finally, the orientation o_v is assigned as e_1, i.e., the eigenvector associated with the largest eigenvalue of the scatter matrix $\mathbf{S}_{3\times 3}$.

Finally, the scale s_v is defined as the longest Euclidean distance between the centers of the central supervoxel and its first-order neighbors in feature region v.

D. Contextual Feature Group Construction

Spatial contextual information exhibits richer, more salient, and distinctive representations than only using local feature regions. Thus, in this paper, to take advantage of spatial contextual information, we construct a contextual feature group for each of the feature regions. Generally, the following four factors should be properly considered for constructing contextual feature groups [63]: 1) the contextual feature group should be scale and rotation invariant; 2) the contextual feature group should be repeatable; 3) the number of feature regions in each contextual feature group should be small; and 4) the construction of contextual feature groups should be computationally efficient. As for the second and third requirements, according to [63] and [64], if too many feature regions are combined, the repeatability of the combination might decrease. In addition, a large contextual feature group with too many feature regions will produce more feature-to-feature matching orders between two contextual feature groups, thereby leading to high computational burdens for measuring spatial contextual similarities between two contextual feature groups in Section III-E. Therefore, to simultaneously meet the aforementioned four requirements, we fix the maximum number of feature regions in each contextual feature group as three. To make a tradeoff between efficiency and effectiveness, for each feature region centered at v, we construct its corresponding contextual feature group $G_v = \{P_v, P_1, P_2, \ldots, P_k\}$ by including feature region v and its k-nearest neighboring feature regions. Fig. 3(b) shows an illustration of the constructed contextual feature group with $k = 2$.

E. Discriminant Group Distance Formulation

Rather than a single feature, a contextual feature group contains several spatial context-related features. Thus, proper metrics should be formulated in order to measure the distance between two contextual feature groups. In this paper, we propose a discriminant group distance, which measures the spatial context weighted *Mahalanobis* distance based on the features in two contextual feature groups. In general, with m features in each contextual feature group, a total number of $m!$ possible feature-to-feature matches exist between two contextual feature groups. Here, we term each possible match as a matching order. For instance, for an $m = 3$, there are 6 matching orders between two contextual feature groups. Thus, the best matching order that maximizes the spatial contextual similarity between two contextual feature groups is used to compute the discriminant group distance.

First, we define the spatial context of each contextual feature group as the orientation and scale relationships between the features within the group. Then, the spatial contextual similarity between two contextual feature groups G_i and G_j is defined as follows:

$$\mathrm{SCSim}(G_i, G_j) = \max_{\psi} \frac{1}{2}\left(\mathrm{OSim}_{\psi}^{(G_i,G_j)} + \mathrm{SSim}_{\psi}^{(G_i,G_j)}\right) \tag{11}$$

where ψ is a matching order; $\mathrm{OSim}_{\psi}^{(G_i,G_j)}$ and $\mathrm{SSim}_{\psi}^{(G_i,G_j)}$ are the orientation and scale similarities under the matching order ψ, respectively. Before giving the definitions of $\mathrm{OSim}_{\psi}^{(G_i,G_j)}$ and $\mathrm{SSim}_{\psi}^{(G_i,G_j)}$, we first define the spatial orientation and scale relationships contained within a contextual feature group G_i under a matching order ψ as follows:

$$\mathrm{ORel}_{\psi}^{(G_i)} = \sum_{x=1,y>x}^{m} \arccos({o_x}^T \cdot o_y) \tag{12}$$

$$\mathrm{SRel}_{\psi}^{(G_i)} = \sum_{x=1,y>x}^{m} \log\left(1 + \frac{s_x}{s_y}\right) \tag{13}$$

where $\mathrm{ORel}_{\psi}^{(G_i)}$ and $\mathrm{SRel}_{\psi}^{(G_i)}$ are the spatial orientation and scale relationships, respectively; m is the number of features in G_i; o_x and o_y are the orientations of feature regions x and y in G_i, respectively; s_x and s_y are the scales of feature regions x and y in G_i, respectively. In fact, $\mathrm{ORel}_{\psi}^{(G_i)}$ and $\mathrm{SRel}_{\psi}^{(G_i)}$ are defined based on the relative orientation difference and scale ratio, respectively. Thus, they are obviously scale and rotation invariant.

Then, based on the definitions of $\mathrm{ORel}_{\psi}^{(G_i)}$ and $\mathrm{SRel}_{\psi}^{(G_i)}$, we define $\mathrm{OSim}_{\psi}^{(G_i,G_j)}$ and $\mathrm{SSim}_{\psi}^{(G_i,G_j)}$ as follows:

$$\mathrm{OSim}_{\psi}^{(G_i,G_j)} = \frac{\min\left(\mathrm{ORel}_{\psi}^{(G_i)}, \mathrm{ORel}_{\psi}^{(G_j)}\right)}{\max\left(\mathrm{ORel}_{\psi}^{(G_i)}, \mathrm{ORel}_{\psi}^{(G_j)}\right)} \tag{14}$$

$$\mathrm{SSim}_{\psi}^{(G_i,G_j)} = \frac{\min\left(\mathrm{SRel}_{\psi}^{(G_i)}, \mathrm{SRel}_{\psi}^{(G_j)}\right)}{\max\left(\mathrm{SRel}_{\psi}^{(G_i)}, \mathrm{SRel}_{\psi}^{(G_j)}\right)}. \tag{15}$$

After computing the orientation and scale similarities under all possible matching orders, we finally obtain the spatial contextual similarity between contextual feature groups G_i and G_j based on (11). Here, we denote the corresponding best matching order as ψ^*, where $\psi^*(x)$ represents the feature in G_j that matches feature x in G_i.

Since each contextual feature group contains both appearance (i.e., a group of 20-D feature descriptors) and spatial contextual information (i.e., orientations and scales), the selected discriminant group distance should properly combine the appearance and spatial contextual properties. To this end, the discriminant group distance between two contextual feature groups G_i and G_j is formulated through a spatial context weighted *Mahalanobis* distance as follows:

$$\begin{aligned}\mathrm{DGDis}(G_i, G_j) = (1 - \mathrm{SCSim}(G_i, G_j)) \\ \cdot \sum_{x=1}^{m} \left(f_x^{G_i} - f_{\psi^*(x)}^{G_j}\right)^T \mathbf{A}^{-1} \left(f_x^{G_i} - f_{\psi^*(x)}^{G_j}\right)\end{aligned} \tag{16}$$

where $\mathbf{A} \in R^{20\times 20}$ is the covariance matrix over all features.

F. Contextual Visual Vocabulary Generation

To generate the contextual visual vocabulary, we first vector-quantize the contextual feature groups into a number of clusters. In our implementation, the vector quantization of contextual feature groups is carried out using k-means clustering based on the discriminant group distance defined in (16). However, in k-means clustering, to update the cluster centers of a cluster C with the defined distance metric, we should properly solve the following optimization problem:

$$G^* = \arg\min_G \sum_{G_j \in \text{ cluster } C} \mathrm{DGDis}(G_j, G). \tag{17}$$

However, it is very time consuming to solve the above problem in each iteration of the k-means clustering. Thus, to make a tradeoff between efficiency and effectiveness, we simply treat the contextual feature group with the maximum similarities to the other members in the same cluster as the updated center:

$$G^* = \arg\min_{G_i} \sum_{G_i, G_j \in \text{ cluster } C, i \neq j} \mathrm{DGDis}(G_j, G_i). \tag{18}$$

To further improve computational efficiency, in practice, we store a group-to-group similarity matrix for each cluster in order to rapidly update the cluster center. Once the similarity matrix of a cluster is computed, the clustering operation in its corresponding sub-clusters can be accomplished efficiently. As shown in Fig. 4, after vector quantization, each cluster center is taken as a distinctive contextual visual word. Finally, such contextual visual words form a contextual visual vocabulary. Since the contextual visual vocabulary is generated using contextual feature groups rather than single feature regions, each word in the vocabulary preserves rich, salient, and distinctive spatial contextual information. In addition, a stop list analogy [65] is used to discard the most frequent contextual visual words that occur in almost all scenes.

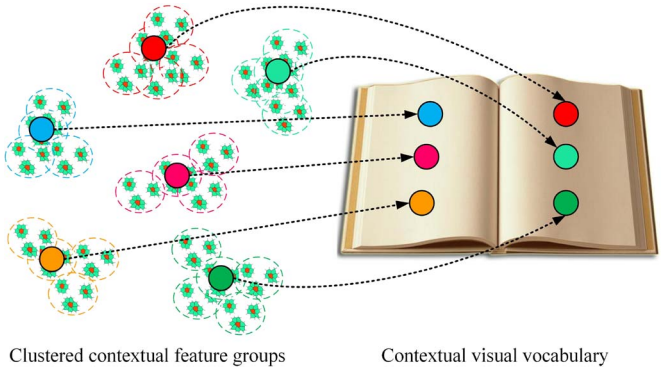


Fig. 4. Contextual visual vocabulary generation.

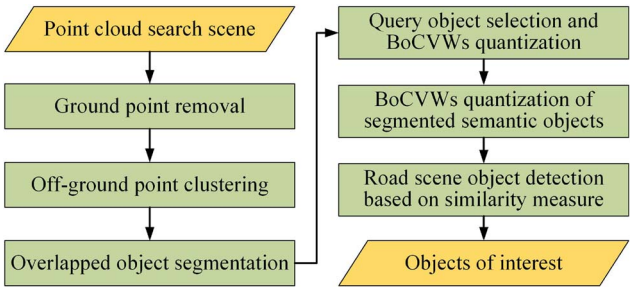


Fig. 5. Road scene object detection workflow.

IV. Road Scene Object Detection

In this section, we present a road scene object detection framework by using the generated contextual visual vocabulary. As shown in Fig. 5, for a search scene, after ground point removal, semantic objects are first segmented through a combination of Euclidean distance clustering and extended voxel-based normalized cut segmentation. Then, the query object and each of the segmented semantic objects are supervoxelized, featured, and quantized to form BoCVWs representations. Finally, the objects of interest are detected based on the similarity measures between the BoCVWs of the query object and the segmented semantic objects.

A. Semantic Object Segmentation

For a search scene, a preprocessing is first performed to remove ground points from the scene using the voxel-based upward growing filtering method [17].

To group the discrete, unorganized off-ground points into semantic objects, first, we apply a Euclidean distance clustering method [17] to the off-ground points to partition them into isolated clusters. Euclidean distance clustering, a nearest neighboring clustering approach, considers the relative Euclidean distances between adjacent points to conduct clustering. Theoretically, an unclustered point was assigned to a specific cluster if and only if its shortest Euclidean distance to the points in this cluster lies below a clustering distance d_c. Otherwise, a new cluster is formed to include this point. Fig. 6(a) shows the off-ground point clustering results using the Euclidean distance clustering method with a clustering distance $d_c = 0.15$ m. In Fig. 6(a), different colors represent different clusters. However, as shown by the two clusters in the black boxes, the overlapped objects cannot be separated by the Euclidean distance clustering method. Therefore, effective means should be developed to further segment such clusters containing multiple overlapped objects.

In our previous study, we propose a voxel-based normalized cut segmentation method [17], which can effectively segment connected and not seriously overlapped objects. However, by considering only geometric features of voxels, this method cannot achieve promising segmentation performance when dealing with seriously overlapped objects. To improve segmentation performance on clusters containing seriously overlapped objects, in this paper, we develop an extended voxel-based normalized cut segmentation method, which integrates intensity features of voxels. As shown in Fig. 7(a), generally, different objects show different retro-reflectivities to near-infrared laser spectra. Such retro-reflectivity difference is reflected by the backscattered intensity information of the laser points in the MLS point clouds. Therefore, intensity information is very useful for semantic object segmentation.

First, the clusters containing multiple overlapped objects are voxelized using the octree partition strategy with a voxel resolution w_s [see Fig. 7(b)]. Then, the generated voxels are organized into a complete weighted graph $\boldsymbol{G} = \{\boldsymbol{V}, \boldsymbol{E}\}$, where the vertices $\boldsymbol{V}$ are formed by the voxels, and the edges $\boldsymbol{E}$ are connected between each pair of voxels. The weights on the edges are used to measure the similarities between the two connected voxels. Such a weight is computed based on the geometric and intensity features of the associated voxels as follows:

$$w_{ij} = \begin{cases} \exp\left(-\frac{\left\|p_i^H - p_j^H\right\|_2^2}{\sigma_H^2}\right) \cdot \exp\left(-\frac{\left|p_i^V - p_j^V\right|^2}{\sigma_V^2}\right) \\ \quad \cdot \exp\left(-\frac{\left|I_i^n - I_j^n\right|^2}{\sigma_I^2}\right), & \text{if } \left\|p_i^H - p_j^H\right\|_2 \le d_H \\ 0, & \text{otherwise.} \end{cases} \tag{19}$$

where w_{ij} is the weight on the edge connecting voxels i and j; $p_i = (x_i, y_i, z_i)$ and $p_j = (x_j, y_j, z_j)$ are the centroids of voxels i and j, respectively. The centroid of a voxel can be computed using (4). $p_i^H = (x_i, y_i)$ and $p_j^H = (x_j, y_j)$ are the coordinates of the centroids on the XY plane; $p_i^V = z_i$ and $p_j^V = z_j$ are the z coordinates of the centroids; I_i^n and I_j^n are the interpolated normalized intensities of the points in voxels i and j, respectively. The interpolated normalized intensity of a voxel can be computed using (8). σ_H^2, σ_V^2, and σ_I^2 are the variances of the horizontal, vertical, and intensity distributions, respectively. d_H is a distance threshold restraining the maximum valid horizontal distance between two voxels. Thus, if the horizontal distance between two voxels exceeds d_H, the weight on the edge connecting these two voxels is set to zero.

In the standard normalized cut segmentation method [66], the cost function for partitioning graph $\boldsymbol{G}$ into two disjoint voxel groups $\boldsymbol{A}$ and $\boldsymbol{B}$ by maximizing the similarity within each voxel group and minimizing the similarity between two voxel groups is defined as follows:

$$\mathrm{Ncut}(\boldsymbol{A}, \boldsymbol{B}) = \frac{\mathrm{cut}(\boldsymbol{A}, \boldsymbol{B})}{\mathrm{assoc}(\boldsymbol{A}, \boldsymbol{V})} + \frac{\mathrm{cut}(\boldsymbol{A}, \boldsymbol{B})}{\mathrm{assoc}(\boldsymbol{B}, \boldsymbol{V})} \tag{20}$$

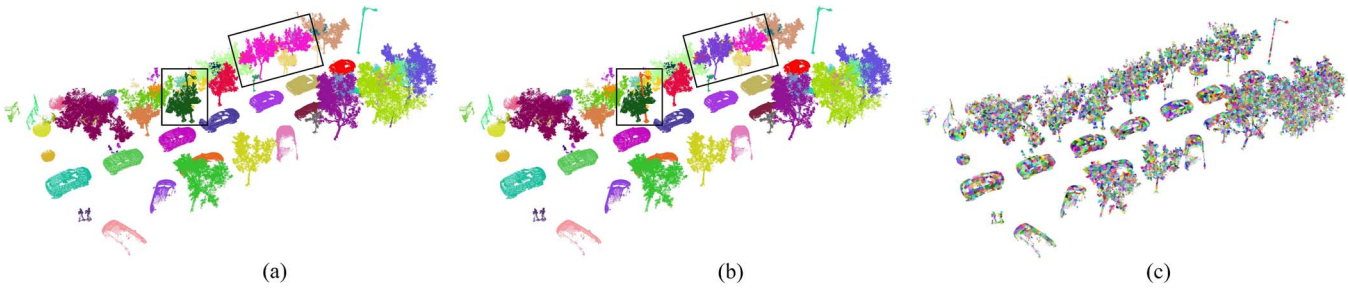


Fig. 6. (a) Clustering results after Euclidean distance clustering. (b) Semantic object segmentation results after extended voxel-based normalized cut segmentation. (c) Supervoxelization results of the semantic objects.

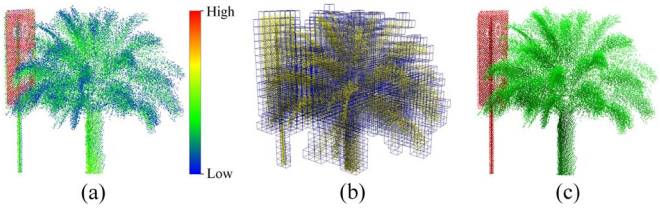


Fig. 7. (a) Cluster containing two overlapped objects rendered with intensity information. (b) Voxelization of the cluster by using the octree partition strategy. (c) Bipartition result of the cluster.

where $\text{cut}(\boldsymbol{A}, \boldsymbol{B}) = \sum_{i \in \boldsymbol{A}, j \in \boldsymbol{B}} w_{ij}$ is the total sum of weights between voxel groups $\boldsymbol{A}$ and $\boldsymbol{B}$; $\text{assoc}(\boldsymbol{A}, \boldsymbol{V}) = \sum_{i \in \boldsymbol{A}, j \in \boldsymbol{V}} w_{ij}$ is the total sum of weights of all edges ending in voxel group $\boldsymbol{A}$. The minimization of $\text{Ncut}(\boldsymbol{A}, \boldsymbol{B})$ is accomplished by solving a corresponding generalized eigenvalue problem [66]. After eigenvalue decomposition, graph $\boldsymbol{G}$ can be bipartitioned into voxel groups $\boldsymbol{A}$ and $\boldsymbol{B}$ by applying a threshold to the eigenvector associated with the second smallest eigenvalue [see Fig. 7(c)]. Fig. 6(b) shows the semantic object segmentation results by using the proposed extended voxel-based normalized cut segmentation method. As shown by the clusters in the black boxes, such clusters containing multiple overlapped objects are well segmented into disjoint semantic objects.

B. Bag-of-Contextual-Visual-Words Quantization

Before carrying out object detection from the segmented off-ground semantic objects, we perform a vector quantization on a 3-D point cloud object to create a bag-of-contextual-visual-words (BoCVWs) representation based on the generated contextual visual vocabulary in Section III. To this end, first, for a 3-D point cloud object, we over-segment it into a supervoxel structure using the VCCS algorithm, and an adjacency graph is constructed to represent adjacency relationships of the supervoxels (see Section III-B). Fig. 8 shows three typical point cloud objects and their corresponding supervoxelization results using the VCCS algorithm. Then, for each supervoxel on the point cloud object, a feature region is constructed by including this supervoxel and its first-order neighbors on the adjacency graph. These feature regions are characterized using our proposed structural-spectral descriptors (see Section III-C). Next, to take advantage of spatial contextual information, for each feature region, a contextual feature group is constructed by integrating this feature region and its k-nearest neighboring

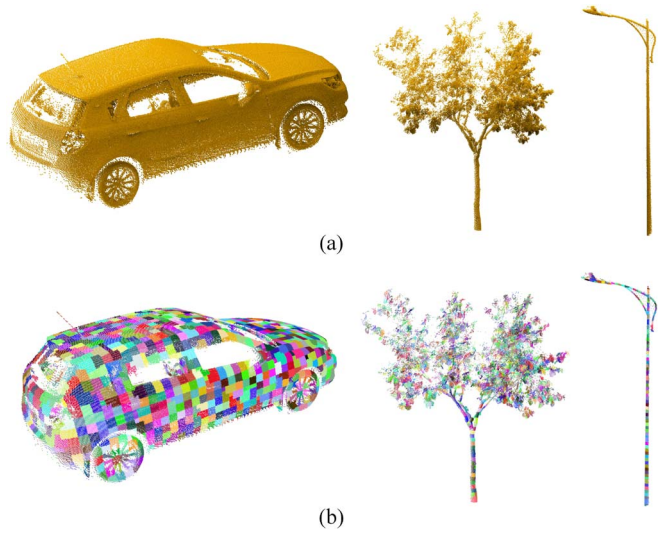


Fig. 8. (a) Three typical 3-D point cloud objects. (b) Corresponding supervoxelization results.

feature regions (see Section III-D). Finally, a contextual visual word is assigned to each contextual feature group. This vector quantization is achieved by ascertaining the nearest cluster center (i.e., the most similar contextual visual word) in the contextual visual vocabulary to the contextual feature group under the discriminant group distance metric.

After vector quantization, a 3-D point cloud object is composed of a set of contextual visual words, each of which encodes a distinctive feature on the point cloud object. Then, we organize such a set of contextual visual words into a BoCVWs representation for depicting this point cloud object. In this paper, we adopt the standard "term frequency-inverse document frequency" weighting [65], [67] to construct the BoCVWs.

Here, we denote a 3-D point cloud object as a document. Given a contextual visual vocabulary of V words, each document d is represented by a V-dimensional vector of weighted word frequencies:

$$v_d = (t_1, t_2, \ldots, t_i, \ldots, t_V)^T \tag{21}$$

where t_i denotes the term frequency-inverse document frequency of the ith word in the vocabulary in document d, and it takes the following form:

$$t_i = \frac{n_i^d}{\sum_{j=1}^{V} n_j^d} \log \frac{N}{N_i} \tag{22}$$

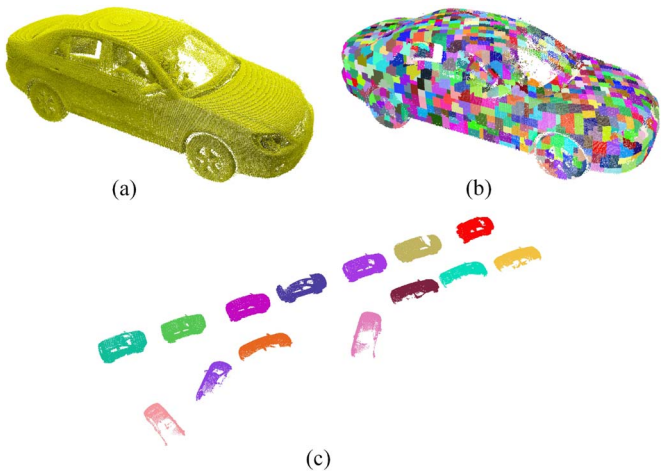


Fig. 9. (a) Query object. (b) Supervoxelization of the query object. (c) Detected objects of interest.

where n_i^d is the number of occurrences of the ith word in document d; N is the total number of documents in the database; N_i is the number of documents containing word i. This weighting is a product of two terms: the *word frequency* and the *inverse document frequency*. Intuitively, the word frequency well weights the words occurring more often in a particular document; whereas the inverse document frequency downweights the words appearing often in the database, thereby improving the distinctiveness of different documents.

In this way, a 3-D point cloud object is represented by a V-dimensional BoCVWs for depicting its unique, distinctive features. This representation is used for road scene object detection in the following section.

C. Road Scene Object Detection

In this section, we conduct object detection from the segmented off-ground semantic objects based on the BoCVWs representations in Section IV-B. To detect a specific category of objects, first, a clean and completely scanned query object is manually selected from the collected point cloud data [see Fig. 9(a)]. Then, the query object and each of the semantic objects in the search scene are supervoxelized, characterized, and quantized to form BoCVWs. Figs. 6(c) and 9(b) present the supervoxelization results of the query object and the semantic objects in the search scene, respectively. Next, based on the BoCVWs, we use the normalized histogram intersection distance metric [68] to measure the similarity between the query object and a semantic object. For a query object $\mathcal{Q}$ and a semantic object P, the similarity between them is defined as follows:

$$\mathrm{Sim}(Q,P)=\frac{\sum_{i=1}^{V}\min\left(v_Q^i,v_P^i\right)}{\sum_{i=1}^{V}\max\left(v_Q^i,v_P^i\right)} \tag{23}$$

where v_Q and v_P are the BoCVWs of objects $\mathcal{Q}$ and P, respectively. Consequently, we compute a series of similarity measures between the query object and all the semantic objects in the search scene. Finally, the similarity measures from all semantic objects are thresholded to obtain the objects of interest. Fig. 9(c) shows the detected objects (cars) from the segmented semantic objects in Fig. 6(b).

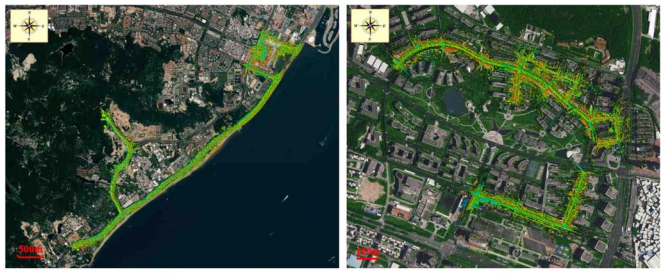


Fig. 10. Surveyed areas and the collected point cloud data sets. (a) Ring Road South data set. (b) Software Park Phase II data set.

V. Results and Discussion

A. MLS Point Cloud Data Sets

In this study, by using the RIEGL VMX-450 MLS system [17], we collected two point cloud data sets on Ring Road South (RRS) and in Software Park Phase II (SPP) in Xiamen City, China (see Fig. 10). The RRS data set contains about 1728 million points and covers a road segment of approximately 11 km. The average point density in this data set is about 4082 points/m^2. This is a typical urban road area containing a great amount of road infrastructure (e.g., light poles and traffic signposts). The SPP data set includes about 626 million points and takes up a road section of approximately 2.5 km. The average point density in this data set is about 4377 points/m^2. This is a typical information technology (IT) development area containing hundreds of IT companies. A performance evaluation on light pole, traffic signpost, and car detection was conducted on the RRS and SPP data sets.

At the contextual visual vocabulary generation stage, we manually, at random, selected a total number of 40 training samples, each of which has a road length of approximately 50 m, from the collected point cloud data.

B. Point Cloud Segmentation

To segment off-ground points into separated semantic objects, we proposed a combination of Euclidean distance clustering and extended voxel-based normalized cut segmentation. Euclidean distance clustering rapidly separates isolated objects; whereas extended voxel-based normalized cut segmentation effectively segments connected or overlapped objects. To evaluate the segmentation performance of our proposed segmentation method, we compared it with the following three segmentation methods: shape-based segmentation method [69], two-step segmentation method [70], and voxel-based normalized cut segmentation method [17]. As shown in Row 1 of Fig. 11, three point cloud scenes were selected for performance comparison. First, the three point cloud scenes were preprocessed to filter out ground points by using the voxel-based upward growing filtering method [17]. The obtained off-ground points after

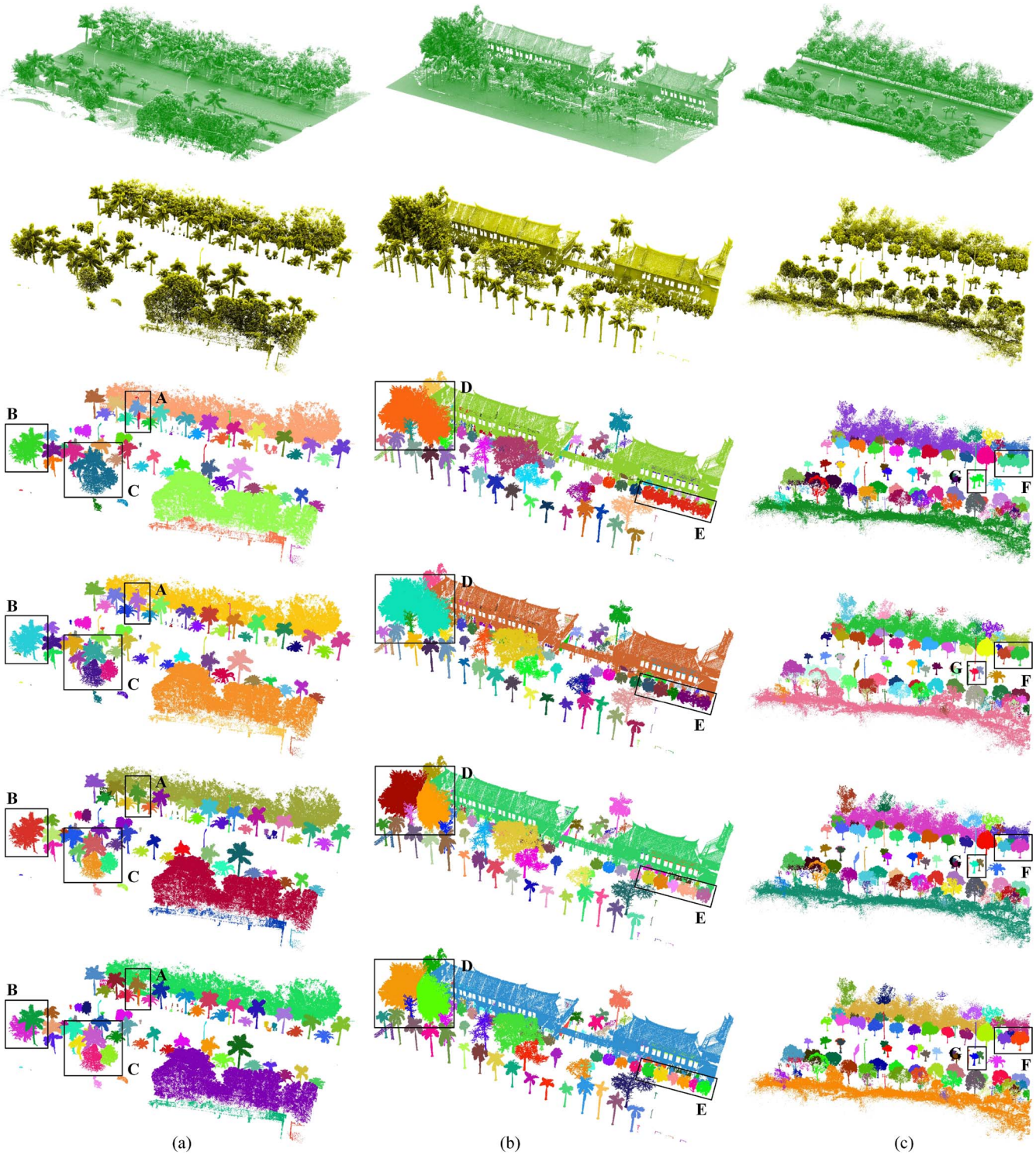


Fig. 11. Point cloud segmentation results on three selected scenes (a), (b), and (c) by using different segmentation methods. Row 1: Raw point clouds of the selected scenes. Row 2: Off-ground points obtained after ground point removal. Row 3: Segmentation results obtained using the shape-based segmentation method [69]. Row 4: segmentation results obtained using the two-step segmentation method [70]. Row 5: segmentation results obtained using the voxel-based normalized cut segmentation method [17]. Row 6: segmentation results obtained using the proposed extended voxel-based normalized cut segmentation method.

ground point removal are shown in Row 2 of Fig. 11. Then, the aforementioned three segmentation methods, as well as our proposed segmentation method, were applied to the off-ground points to perform segmentation. The segmentation results obtained using the shape-based segmentation method, the two-step segmentation method, the voxel-based normalized cut segmentation method, and our proposed segmentation method are shown in Rows 3, 4, 5, and 6, respectively.

On the whole, these four methods all achieved promising segmentation results on the three selected scenes. However, as shown by the boxes labeled C, E, F, and G, these point cloud clusters containing multiple trees of the same category

failed to be segmented by using the shape-based segmentation method; while such clusters were well separated into individual trees by using the other three methods. As shown by the box labeled D, this cluster contains two overlapped kapok trees. The shape-based and the two-step segmentation methods were not able to segment this cluster into two separated kapok trees. On the contrary, the voxel-based normalized cut segmentation method and our proposed segmentation method worked well in segmenting such clusters. Moreover, as shown by the box labeled A, a light pole is hidden in a palm tree and seriously overlapped with the palm tree. Such a cluster failed to be segmented by using the two-step and the voxel-based normalized cut segmentation methods. However, benefited from the use of shape features and intensity features, respectively, the shape-based segmentation method and our proposed segmentation method both obtained superior segmentation performance in dealing with such clusters. In addition, as shown by the box labeled B, a sago cycas tree is connected very closely to a palm tree. Thus, the two-step and the voxel-based normalized cut segmentation methods all failed to segment them. Due to the high similarities between the leaves of the sago cycas tree and the palm tree, the shape-based segmentation method also failed to handle this cluster. Comparatively, by considering retro-reflectivity properties of objects, our proposed segmentation method achieved a promising segmentation result and successfully segmented this cluster into two disjoint components.

In conclusion, the shape-based segmentation method well segments isolated clusters and overlapped clusters containing objects of different categories; however, it lacks of capability to handle clusters containing closely overlapped objects of the same category. The two-step and the voxel-based normalized cut segmentation methods are able to deal with separated and not seriously overlapped clusters; however, they have problems in segmenting clusters containing seriously overlapped objects from either the same or different categories. Comparatively, our proposed segmentation method obtains relatively better performance than the other three methods and it has the capability of tackling isolated clusters, connected clusters, and even seriously overlapped clusters.

C. Parameter Sensitivity Analysis

In the proposed algorithm, the configurations of the following three parameters have a significant impact on the road scene object detection performance: feature region construction pattern, contextual visual vocabulary size (V), and contextual feature group size $(k+1)$. In order to obtain an optimal configuration for each of these parameters, we conducted a group of experiments to test the performance of each parameter configuration on the road scene object detection results. For feature region construction, we tested the following two construction patterns: using single supervoxels and using the integration of supervoxels and their first-order neighbors. For contextual visual vocabulary generation, we tested the following six configurations: $V = 90{,}000$, 100,000, 110,000, 120,000, 130,000, and 140, 000. For contextual feature group construction, we tested the following six configurations: $k = 0$, 1, 2, 3, 4, and 5. Here, $k = 0$ means that single feature regions

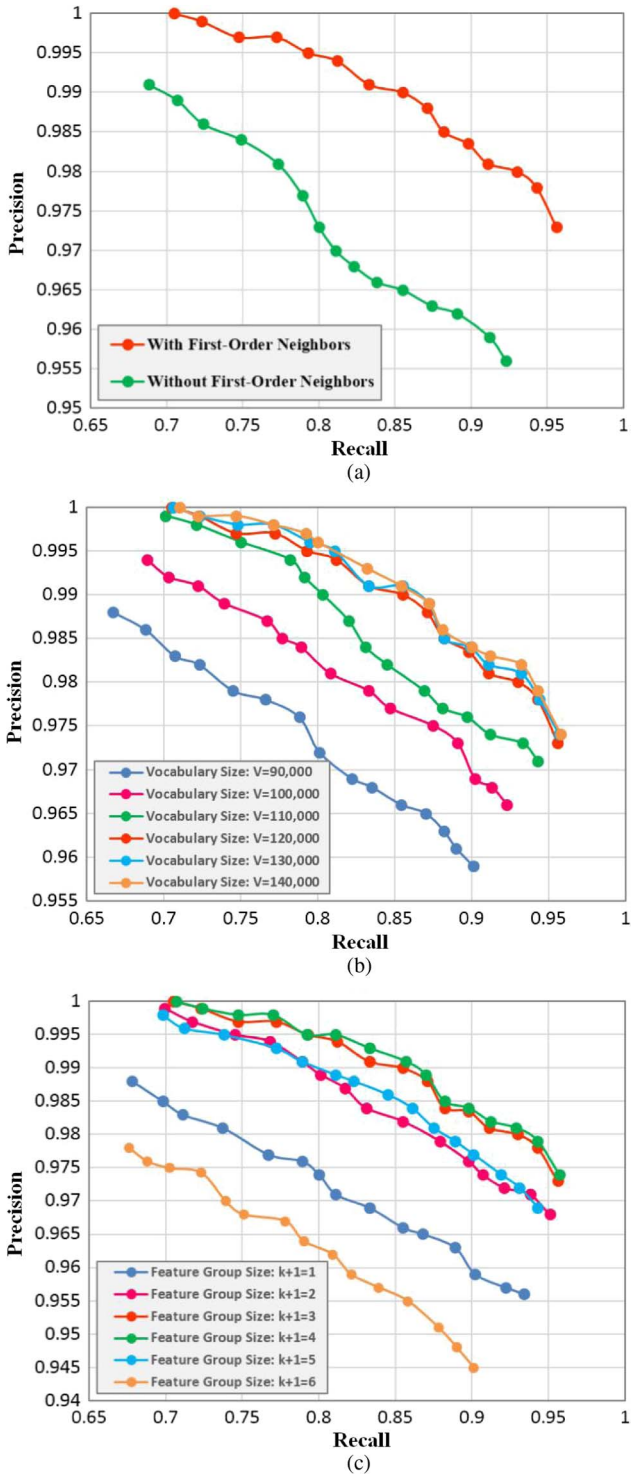


Fig. 12. Performances obtained under different parameter configurations. (a) Feature region construction with and without first-order neighbors. (b) Contextual visual vocabulary size. (c) Contextual feature group size.

(without spatial contextual information) are used to generate the contextual visual vocabulary. The test results of these parameter configurations were presented and analyzed using precision-recall curves (see Fig. 12). As shown in Fig. 12(a), by using first-order neighbors to construct feature regions, the detection performance improves greatly than that of using only single supervoxels as feature regions. This is because feature regions with first-order neighborhood information can produce more

TABLE I
PARAMETERS AND THEIR CONFIGURATIONS

k	V	d_c	w_s	σ_H	σ_V	σ_I	d_H
2	120,000	0.15 m	0.1 m	2 m	10 m	1.0	5 m

salient, distinctive features than that of using only local supervoxels. Thus, we constructed feature regions by integrating first-order neighbors of supervoxels in our experiments. As shown in Fig. 12(b), the detection performance improves as the vocabulary size increases. This is because the more the contextual visual words in the vocabulary, the higher degrees of distinctions between different categories of objects. However, when the vocabulary size exceeds 120,000, the performance changes very slightly. In addition, the increase of the vocabulary size brings great computational burdens at the vocabulary generation stage. Thus, balancing detection performance and computational complexity, we set the vocabulary size at $V = 120{,}000$. As shown in Fig. 12(c), when $k \leq 3$, the detection performance improves with the increase of the contextual feature group size. This is because, by considering spatial contextual information of feature regions, the quantized contextual visual words are more likely to obtain salient, distinctive feature encodings, thereby capable of differentiating objects of different categories. However, when $k \geq 4$, the detection performance drops dramatically. In fact, if too many local feature regions are combined, the repeatability of the combination decreases accordingly, leading to a detection performance decrease. In addition, the increase of k slows down the generation of the contextual visual vocabulary. Therefore, to obtain acceptable detection performance, we set the contextual feature group size at 3 (i.e., $k = 2$).

D. Road Scene Object Detection

To evaluate the performance of our proposed road scene object detection algorithm, we applied it to the aforementioned two point cloud data sets (i.e., RRS and SPP data sets). We respectively conducted light pole, traffic signpost, and car detection on the RRS and SPP data sets. After parameter sensitivity analysis, the optimal parameter configurations used in the proposed algorithm are detailed in Table I. To improve computational efficiency, first, these two data sets were preprocessed to filter out ground points through voxel-based upward growing filtering. Then, the remaining off-ground points were clustered and segmented into individual semantic objects via Euclidean distance clustering and extended voxel-based normalized cut segmentation. To detect the objects of interest, a group of query objects were selected from the collected point clouds. Next, the query objects and each of the segmented semantic objects were supervoxelized, featured, and quantized to form a set of BoCVWs. Finally, the objects of interest were detected by comparing the similarities between the query objects and the semantic objects based on the BoCVWs.

The road scene object detection results, along with the manually labeled ground truth, on the two selected data sets are shown in Table II. Fig. 13 shows two visual examples of parts of the road scene object detection results on the two selected data sets. As reflected in Table II and Fig. 13, the majority of the road scene objects were correctly detected, including different shapes of traffic signposts, light poles with and without attachments (e.g., traffic signs and advertising boards), and cars of different levels of completeness. However, as shown by the object labeled A in Fig. 13(a), due to the high geometric similarity of the traffic camera to the light pole, this traffic camera was falsely detected as a light pole by using the proposed algorithm. Moreover, as shown by the object labeled B in Fig. 13(b), due to occlusions, a car was scanned with very bad data coverage; thus, this car failed to be detected because of insufficient features. In addition, as shown by the objects labeled C and D in Fig. 13(b), caused by the Doppler Effect, two moving cars were seriously distorted in the resultant point cloud data; therefore, these two cars also failed to be detected because of high geometric dissimilarities. On the whole, the proposed algorithm obtained very promising performance in detecting light poles, traffic signposts, and cars from large-volume point cloud data.

To quantitatively assess the accuracy of the road scene object detection results, we adopted the following four indices: recall, precision, quality, and F-score [17], [20]. Recall evaluates the ratio of the correctly detected objects to the ground

TABLE II
GROUND TRUTH AND ROAD SCENE OBJECT DETECTION RESULTS

Data set	Ground Truth			Detection Results			
	Light Pole	Traffic Signpost	Car	Light Pole	Traffic Signpost	Car	False Positive
RRS	647	241	159	628	235	143	31
SPP	167	25	780	158	22	731	29

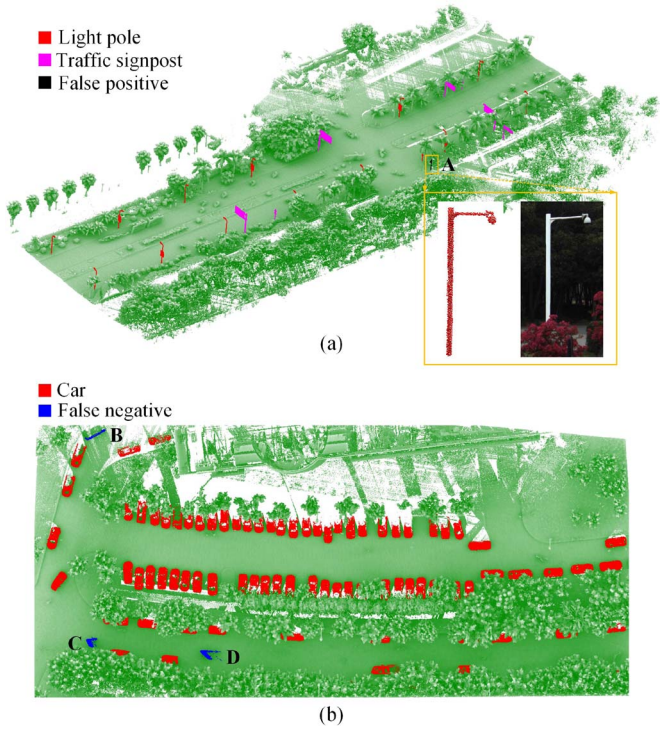


Fig. 13. Parts of the road scene object detection results. (a) Light pole and traffic signpost detection results on the RRS data set. (b) Car detection results on the SPP data set.

TABLE III
QUANTITATIVE EVALUATION RESULTS

Data set	Recall	Precision	Quality	F-score
RRS	0.961	0.970	0.933	0.965
SPP	0.937	0.969	0.910	0.953
Average	0.949	0.970	0.922	0.959

TABLE IV
COMPUTING TIME ON THE TWO SELECTED DATA SETS (UNIT: SECOND)

Dataset	Ground Removal	Object Segmentation	BoCVWs Quantization	Object Detection	Total
RRS	31	133	2318	29	2511
SPP	12	36	1585	17	1650

truth; precision measures the ratio of the correctly detected objects to all the detected components; quality and F-score are two overall measures. The four indices are defined as follows: recall $= TP/(TP + FN)$, precision $= TP/(TP + FP)$, quality $= TP/(TP + FN + FP)$, and F-score $= 2 \times$ recall $\times$ precision$/$(recall $+$ precision), where TP, FN, and FP denote the numbers of true positives, false negatives, and false positives, respectively. The quantitative evaluation results using these four indices are detailed in Table III. The proposed algorithm achieved an average recall, precision, quality, and F-score of 0.949, 0.970, 0.922, and 0.959, respectively, in detecting light poles, traffic signposts, and cars on the two selected data sets. Therefore, the proposed algorithm performs efficiently in detecting road scene objects from MLS point cloud data.

The proposed algorithm was implemented using C++ and tested on an HP Z820 8-core-16-thread workstation. The computing times at the contextual visual vocabulary generation and the road scene object detection stages were recorded for time complexity analysis. The total computing time for generating the contextual visual vocabulary containing 120,000 contextual visual words was approximately 39 min. At the road scene object detection stage, each data set was first partitioned into a group of data segments with a road length of about 50 m. Then, all the segments were fed into a multithread computing environment containing 16 parallel threads. Such a parallel computing strategy dramatically improves the computational efficiency and reduces the time complexity of the proposed algorithm. The detailed computing time in each processing step is listed in Table IV. As reflected in Table IV, the total time cost for detecting light poles, traffic signposts, and cars on the RRS data set was about 42 min; the total processing time for detecting light poles, traffic signposts, and cars on the SPP data set was about 28 min. Therefore, the proposed algorithm is suitable for rapidly handling large-volume MLS point clouds toward road scene object detection.

E. Comparative Studies

Comparative studies were also conducted to further compare the detection performance between our proposed algorithm and the following three existing methods: the Hough forest-based method (HF) [30], the 3-D object matching-based method (OM) [43], and the bottom-up and top-down descriptors-based method (BTD) [46]. A performance evaluation on car detection was conducted on the RRS and SPP data sets, as well as the publicly available Paris-Rue-Madame data set (PRM) [71], by using the above three methods, as well as our proposed algorithm. The ground truths of cars are 159, 780, and 70, respectively, in the RRS, SPP, and PRM data sets. The car detection results and quantitative evaluation results obtained by using these three methods, as well as our proposed algorithm, are listed in Table V. The HF method used a pre-trained part-based Hough forest model to detect cars. However, in the selected data sets, some cars were scanned with serious incompleteness; thus, such cars failed to be detected by using the HF method. Moreover, the OM method relied greatly on the off-ground semantic object segmentation results. However, in the selected data sets, some cars are hidden in the trees and overlapped seriously with the trees. The voxel-based normalized cut segmentation method proposed in this method could not effectively segment such clusters into individual components; thus, such cars failed to be detected by using the OM method. The BTD method performed well on the cars with good completeness. However, it also failed to detect cars scanned with serious incompleteness. Comparatively, benefited from the use of the retro-reflectivity properties of objects in our proposed extended voxel-based normalized cut segmentation method in this paper, our proposed algorithm is able to deal with those cars seriously overlapped with the trees. Therefore, our proposed algorithm attained more true positives and relatively less false positives than the other three methods. However, some moving cars distorted seriously in the selected data sets failed to be detected by using all these four methods. In conclusion, our proposed algorithm outperformed the other three methods in accurately and completely detecting cars from MLS point clouds.

TABLE V
CAR DETECTION RESULTS OBTAINED BY USING DIFFERENT METHODS

Data set	Method	Detection Result		Quantitative Evaluations			
		TP	FP	Recall	Precision	Quality	F-score
RRS	HF [30]	138	10	0.868	0.932	0.817	0.899
	OM [43]	141	8	0.887	0.946	0.844	0.916
	BTD [46]	139	7	0.874	0.952	0.837	0.911
	Proposed	143	8	0.899	0.947	0.856	0.922
SPP	HF [30]	711	25	0.912	0.966	0.883	0.938
	OM [43]	723	20	0.927	0.973	0.904	0.949
	BTD [46]	718	23	0.921	0.969	0.894	0.944
	Proposed	731	21	0.937	0.972	0.913	0.954
PRM [71]	HF [30]	61	6	0.871	0.910	0.803	0.890
	OM [43]	65	5	0.929	0.929	0.867	0.929
	BTD [46]	63	4	0.900	0.940	0.851	0.920
	Proposed	66	4	0.943	0.943	0.892	0.943

VI. CONCLUSION

In this paper, we have presented a novel algorithm for detecting road scene objects in MLS data based on BoCVWs.

The proposed algorithm was evaluated on two point cloud data sets for detecting light poles, traffic signposts, and cars directly from large-volume 3-D MLS point cloud data. Quantitative evaluations showed that the proposed algorithm achieved an average recall, precision, quality, and F-score of 0.949, 0.970, 0.922, and 0.959, respectively, in detecting light poles, traffic signposts, and cars on the two selected data sets. Through computational efficiency analysis, by adopting a multithread computing strategy, the proposed algorithm can rapidly handle large-volume MLS point clouds toward road scene object detection. In addition, comparative studies also demonstrated that the proposed algorithm outperformed the other three existing methods in accurately and completely detecting cars of varying conditions. In conclusion, by using MLS point cloud data, we have provided a promising and effective solution to rapid, accurate detection of road scene objects toward transportation-related applications.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments.

REFERENCES

[1] N. Zheng *et al.*, "Toward intelligent driver-assistance and safety warning systems," *IEEE Intell. Syst.*, vol. 19, no. 2, pp. 8–11, Mar./Apr. 2004.

[2] H. Cheng, N. Zheng, X. Zhang, J. Qin, and H. van de Wetering, "Interactive road situation analysis for driver assistance and safety warning systems: Framework and algorithms," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 1, pp. 157–167, Mar. 2007.

[3] J. Choi *et al.*, "Environment-detection-and-mapping algorithm for autonomous driving in rural or off-road environment," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 2, pp. 974–982, Jun. 2012.

[4] A. Broggi *et al.*, "Extensive tests of autonomous driving technologies," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1403–1415, Sep. 2013.

[5] Y. W. Seo, J. Lee, W. Zhang, and D. Wettergreen, "Recognition of highway workzones for reliable autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 708–718, Apr. 2015.

[6] Y. Lv, Y. Duan, W. Kang, Z. Li, and F. Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, Apr. 2015.

[7] A. Abadi, T. Rajabioun, and P. A. Ioannou, "Traffic flow prediction for road transportation networks with limited traffic data," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 653–662, Apr. 2015.

[8] Y. Xu, Q. J. Kong, R. Klette, and Y. Liu, "Accurate and interpretable Bayesian MARS for traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 6, pp. 2457–2469, Dec. 2014.

[9] S. Murray *et al.*, "Mobile mapping system for the automated detection and analysis of road delineation," *IET Intell. Transp. Syst.*, vol. 5, no. 4, pp. 221–230, Dec. 2011.

[10] M. Brogan, S. McLoughlin, and C. Deegan, "Assessment of stereo camera calibration techniques for a portable mobile mapping system," *IET Comput. Vis.*, vol. 7, no. 3, pp. 209–217, Jun. 2013.

[11] K. Williams, M. J. Olsen, G. V. Roe, and C. Glennie, "Synthesis of transportation applications of mobile LiDAR," *Remote Sens.*, vol. 5, no. 9, pp. 4652–4692, Sep. 2013.

[12] C. K. Toth, "R&D of mobile LiDAR mapping and future trends," in *Proc. ASPRS Annu. Conf.*, Baltimore, MD, USA, 2009, pp. 1–7.

[13] J. Gong, H. Zhou, C. Gordon, and M. Jalayer, "Mobile terrestrial laser scanning for highway inventory data collection," in *Proc. Int. Conf. Comput. Civil Eng.*, Clearwater Beach, FL, USA, 2012, pp. 17–20.

[14] H. Guan *et al.*, "Iterative tensor voting for pavement crack extraction using mobile laser scanning data," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 3, pp. 1527–1537, Mar. 2015.

[15] Y. Yu, H. Guan, and Z. Ji, "Automated detection of urban road manhole covers using mobile laser scanning data," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 3258–3269, Dec. 2015.

[16] S. Pu, M. Rutzinger, G. Vosselman, and S. O. Elberink, "Recognizing basic structures from mobile laser scanning data for road inventory studies," *ISPRS J. Photogramm. Remote Sens.*, vol. 66, no. 6, pp. S28–S39, Dec. 2011.

[17] Y. Yu, J. Li, H. Guan, and C. Wang, "Automated extraction of urban road facilities using mobile laser scanning data," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 2167–2181, Aug. 2015.

[18] 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.

[19] S. I. El-Halawany and D. D. Lichti, "Detection of road poles from mobile terrestrial laser scanner point cloud," in *Proc. Int. Workshop Multi-Platform/Multi-Sensor Remote Sens. Mapping*, Xiamen, China, 2011, pp. 1–6.

[20] 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.

[21] C. Cabo, C. Ordoñez, S. García-Cortés, and J. Martínez, "An algorithm for automatic detection of pole-like street furniture objects from mobile laser scanning point clouds," *ISPRS J. Photogramm. Remote Sens.*, vol. 87, pp. 47–56, Jan. 2014.

[22] S. I. El-Halawany and D. D. Lichti, "Detecting road poles from mobile terrestrial laser scanning data," *GISci. Remote Sens.*, vol. 50, no. 6, pp. 704–722, Dec. 2013.

[23] Y. Chen, H. Zhao, and R. Shibasaki, "A mobile system combining laser scanners and cameras for urban spatial objects extraction," in *Proc. IEEE Conf. Mach. Learn. Cybern.*, Pokfulam, Hong Kong, 2007, vol. 3, pp. 1729–1733.

[24] Y. Hu, X. Li, J. Xie, and L. Guo, "A novel approach to extracting street lamps from vehicle-borne laser data," in *Proc. IEEE Conf. Geoinf.*, Shanghai, China, 2011, pp. 1–6.

[25] M. Lehtomäki, A. Jaakkola, J. Hyyppä, 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, Feb. 2010.

[26] A. Kukko, A. Jaakkola, M. Lehtomäki, H. Kaartinen, and Y. Chen, "Mobile mapping system and computing methods for modeling of road environment," in *Proc. Urban Remote Sens. Event*, Shanghai, China, 2009, pp. 1–6.

[27] D. Manandhai and R. Shibasaki, "Feature extraction from range data," in *Proc. Asian Conf. Remote Sens.*, Singapore, 2001, vol. 5, pp. 1–6.

[28] H. Yokoyama, H. Date, S. Kanai, and H. Takeda, "Pole-like objects recognition from mobile laser scanning data using smoothing and principal component analysis," in *Proc. Int. Archives Photogramm. Remote Sens. Spatial Inf. Sci.*, 2011, vol. 38-5/W12, pp. 1–6.

[29] H. Wang *et al.*, "Object detection in terrestrial laser scanning point clouds based on Hough forest," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 10, pp. 1807–1811, Oct. 2014.

[30] H. Wang *et al.*, "3-D point cloud object detection based on supervoxel neighborhood with Hough forest framework," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 4, pp. 1570–1581, Apr. 2015.

[31] L. Zhou and Z. Deng, "LiDAR and vision-based real-time traffic sign detection and recognition algorithm for intelligent vehicle," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, Qingdao, China, 2014, pp. 578–583.

[32] C. Ai and Y. J. Tsai, "Critical assessment of an enhanced traffic sign detection method using mobile LiDAR and INS technologies," *J. Transp. Eng.*, vol. 141, no. 5, pp. 1–12, May 2015.

[33] C. Wen *et al.*, "Spatial-related traffic sign inspection for inventory purposes using mobile laser scanning data," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 1, pp. 27–37, Jan. 2016.

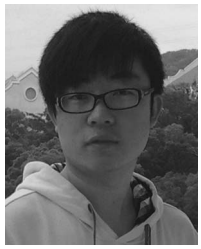
[34] X. Chen *et al.*, "Next generation map making: Geo-referenced ground-level LiDAR point clouds for automatic retro-reflective road feature extraction," in *Proc. ACM SIGSPATIAL Int. Conf. Adv. Geograph. Inf. Syst.*, Seattle, WA, USA, 2009, pp. 488–491.

[35] A. Vu, Q. Yang, J. A. Farrell, and M. Barth, "Traffic sign detection, state estimation, and identification using onboard sensors," in *Proc. IEEE Int. Annu. Conf. Intell. Transp. Syst.*, Hague, The Netherlands, 2013, pp. 875–880.

[36] A. Börcs and C. Benedek, "A marked point process model for vehicle detection in aerial LiDAR point clouds," in *Proc. ISPRS Annu. Photogramm. Remote Sens. Spatial Inf. Sci.*, Melbourne, Vic., Australia, 2012, vol. 1–3, pp. 93–98.

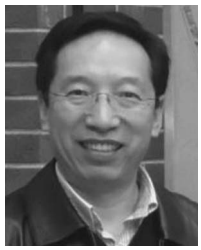
[37] X. Descombes, R. Minlos, and E. Zhizhina, "Object extraction using a stochastic birth-and-death dynamics in continuum," *J. Math. Imag. Vis.*, vol. 33, no. 3, pp. 347–359, Mar. 2009.

[38] A. Börcs and C. Benedek, "Extraction of vehicle groups in airborne LiDAR point clouds with two-level point processes," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 3, pp. 1475–1489, Mar. 2015.
[39] M. Varela-González, H. González-Jorge, B. Riveiro, and P. Arias, "Automatic filtering of vehicles from mobile LiDAR datasets," *Measurement*, vol. 53, pp. 215–223, Jul. 2014.
[40] W. Yao, S. Hinz, and U. Stilla, "Automatic vehicle extraction from airborne LiDAR data of urban areas aided by geodesic morphology," *Pattern Recognit. Lett.*, vol. 31, no. 10, pp. 1100–1108, Jul. 2010.
[41] W. Yao, S. Hinz, and U. Stilla, "Extraction and motion estimation of vehicles in single-pass airborne LiDAR data towards urban traffic analysis," *ISPRS J. Photogramm. Remote Sens.*, vol. 66, no. 3, pp. 260–271, May 2011.
[42] J. Zhang, M. Duan, Q. Yan, and X. Lin, "Automated vehicle extraction from airborne LiDAR data using an object-based point cloud analysis method," *Remote Sens.*, vol. 6, no. 9, pp. 8405–8423, Sep. 2014.
[43] Y. Yu, J. Li, H. Guan, F. Jia, and C. Wang, "Three-dimensional object matching in mobile laser scanning point clouds," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 3, pp. 492–496, Mar. 2015.
[44] B. Fortin, R. Lherbier, and J. C. Noyer, "Feature extraction in scanning laser range data using invariant parameters: Application to vehicle detection," *IEEE Trans. Veh. Tech.*, vol. 61, no. 9, pp. 3838–3850, Nov. 2012.
[45] W. Yao and U. Stilla, "Comparison of two methods for vehicle extraction from airborne LiDAR data toward motion analysis," *IEEE Geosci. Remote Sens. Lett.*, vol. 8, no. 4, pp. 607–611, Jul. 2011.
[46] A. Patterson, IV, P. Mordohai, and K. Daniilidis, "Object detection from large-scale 3-D datasets using bottom-up and top-down descriptors," in *Proc. Eur. Conf. Comput. Vis.*, Marseille, France, 2008, vol. 5305, pp. 553–566.
[47] A. Serna and B. Marcotegui, "Detection, segmentation and classification of 3D urban objects using mathematical morphology and supervised learning," *ISPRS J. Photogramm. Remote Sens.*, vol. 93, pp. 243–255, Jul. 2014.
[48] M. Weinmann, B. Jutzi, S. Hinz, and C. Mallet, "Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers," *ISPRS J. Photogramm. Remote Sens.*, vol. 105, pp. 286–304, Jul. 2015.
[49] Y. Zhou, Y. Yu, G. Lu, and S. Du, "Super-segments based classification of 3D urban street scenes," *Int. J. Adv. Robot. Syst.*, vol. 9, no. 248, pp. 1–8, Oct. 2012.
[50] A. K. Aijazi, P. Checchin, and L. Trassoudaine, "Segmentation based classification of 3D urban point clouds: A super-voxel based approach with evaluation," *Remote Sens.*, vol. 5, no. 4, pp. 1624–1650, Mar. 2013.
[51] H. Zhao, Y. Liu, X. Zhu, Y. Zhao, and H. Zha, "Scene understanding in a large dynamic environment through a laser-based sensing," in *Proc. IEEE Int. Conf. Robot. Autom.*, Anchorage, AK, USA, 2010, pp. 127–133.
[52] S. Friedman and I. Stamos, "Online detection of repeated structures in point clouds of urban scenes for compression and registration," *Int. J. Comput. Vis.*, vol. 102, no. 1, pp. 112–128, Mar. 2013.
[53] B. Douillard, J. Underwood, V. Vlaskine, A. Quadros, and S. Singh, "A pipeline for the segmentation and classification of 3D point clouds," *Experimental Robotics*. Berlin, Germany: Springer-Verlag, 2014, pp. 585–600.
[54] B. Yang, Z. Wei, Q. Li, and J. Li, "Semiautomated building façade footprint extraction from mobile LiDAR point clouds," *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 4, pp. 766–770, Jul. 2013.
[55] A. Jochem, B. Höfle, and M. Rutzinger, "Extraction of vertical walls from mobile laser scanning data for solar potential assessment," *Remote Sens.*, vol. 3, no. 4, pp. 650–667, Mar. 2011.
[56] Y. Yu, J. Li, H. Guan, D. Zai, and C. Wang, "Automated extraction of 3D trees from mobile LiDAR point clouds," in *Proc. Int. Archives Photogramm. Remote Sens. Spatial Inf. Sci.*, 2014, vol. XL-5, pp. 629–632.
[57] H. Guan, Y. Yu, Z. Ji, J. Li, and Q. Zhang, "Deep learning-based tree classification using mobile LiDAR data," *Remote Sens. Lett.*, vol. 6, no. 11, pp. 864–873, Sep. 2015.
[58] L. Cheng, L. Tong, Y. Wang, and M. Li, "Extraction of urban power lines from vehicle-borne LiDAR data," *Remote Sens.*, vol. 6, no. 4, pp. 3302–3320, Apr. 2014.
[59] Y. Yu, J. Li, H. Guan, F. Jia, and C. Wang, "Learning hierarchical features for automated extraction of road markings from 3-D mobile LiDAR point clouds," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 2, pp. 709–726, Feb. 2015.
[60] J. Papon, A. Abramov, M. Schoeler, and F. Wörgötter, "Voxel cloud connectivity segmentation—Supervoxels for point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Portland, OR, USA, 2013, pp. 2017–2034.
[61] B. Yang, Z. Dong, G. Zhao, and W. Dai, "Hierarchical extraction of urban objects from mobile laser scanning data," *ISPRS J. Photogramm. Remote Sens.*, vol. 99, pp. 45–57, Jan. 2015.
[62] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3-D registration," in *Proc. IEEE Int. Conf. Robot. Autom.*, Kobe, Japan, 2009, pp. 3212–3217.
[63] S. Zhang *et al.*, "Building contextual visual vocabulary for large-scale image applications," in *Proc. Int. Conf. Multimedia*, Firenze, Italy, 2010, pp. 501–510.
[64] D. Liu, G. Hua, P. Viola, and T. Chen, "Integrated feature selection and higher-order spatial feature extraction for object categorization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Anchorage, AK, USA, 2008, pp. 1–8.
[65] J. Sivic and A. Zisserman, "Efficient visual search of videos cast as text retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 4, pp. 591–606, Apr. 2009.
[66] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.
[67] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. New York, NY, USA: ACM Press, 1999.
[68] Y. Jiang, J. Meng, J. Yuan, and J. Luo, "Randomized spatial context for object search," *IEEE Trans. Image Process.*, vol. 24, no. 6, pp. 1748–1762, Jun. 2015.
[69] B. Yang and Z. Dong, "A shape-based segmentation method for mobile laser scanning point clouds," *ISPRS J. Photogramm. Remote Sens.*, vol. 81, pp. 19–30, Jul. 2013.
[70] Y. Zhou *et al.*, "A fast and accurate segmentation method for ordered LiDAR point cloud of large-scale scenes," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 11, pp. 1984–1985, Nov. 2014.
[71] A. Serna, B. Marcotegui, F. Goulette, and J. E. Deschaud, "Paris-Rue-Madame database: A 3D mobile laser scanner dataset for benchmarking urban detection, segmentation and classification methods," in *Proc. Int. Conf. Pattern Recog. Appl. Method*, Angers, France, 2014, pp. 1–6.



Yongtao Yu (M'16) received the Ph.D. degree in computer science and technology from Xiamen University, Xiamen, China, in 2015.

He is an Assistant Professor with the Faculty of Computer and Software Engineering, Huaiyin Institute of Technology, Huaian, China. He is also with Fujian Key Laboratory of Sensing and Computing for Smart Cities, Xiamen University. He is the author or coauthor of over 20 research papers published in refereed journals and proceedings. His research interests include pattern recognition, computer vision, machine learning, intelligent interpretation of 3-D point clouds, and remotely sensed imagery.



Jonathan Li (M'00–SM'11) received the Ph.D. degree in geomatics engineering from University of Cape Town, Cape Town, South Africa, in 2000.

He is a Professor with the Department of Geography and Environmental Management, University of Waterloo, Waterloo, ON, Canada, and with Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Information Science and Engineering, Xiamen University, Xiamen, China. He is the author or coauthor of more than 300 publications, over 100 of which are published in refereed journals, including *Remote Sensing of Environment*, IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS (IEEE TITS), IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING (IEEE JSTARS), IEEE GEOSCIENCE AND REMOTE SENSING LETTERS, *International Journal of Remote Sensing*, *Photogrammetric Engineering and Remote Sensing*, and *International Society for Photogrammetry and Remote Sensing (ISPRS) Journal of Photogrammetry and Remote Sensing*. His research interests include mobile laser scanning, point cloud processing, and feature extraction.

Dr. Li serves as the Chair of ISPRS ICWG I/Va on Mobile Scanning and Imaging Systems (2012–2016) and the Chair of ICA Commission on Sensor-Driven Mapping (2015–2019). He also serves as the Associate Editor for IEEE TITS and JSTARS.

Haiyan Guan (M'15) received the Ph.D. degree in geomatics from University of Waterloo, Waterloo, ON, Canada, in 2014.

She is a Professor with the College of Geography and Remote Sensing, Nanjing University of Information Science and Technology, Nanjing, China. She is the author or coauthor of over 30 research papers published in refereed journals and proceedings. Her research interests include airborne, terrestrial, and mobile laser scanning data processing algorithms and 3-D spatial modeling and reconstruction of critical infrastructure and landscape.

Cheng Wang (M'12–SM'16) received the Ph.D. degree in information and communication engineering from National University of Defense Technology, Changsha, China, in 2002.

He is a Professor and the Executive Director of Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Information Science and Engineering, Xiamen University, Xiamen, China. He is the author of almost 100 papers in refereed journals and proceedings. His research interests include remote sensing image processing, mobile laser scanning data analysis, and multisensor fusion.

Dr. Wang is currently the Cochair of International Society for Photogrammetry and Remote Sensing WG I/3 on Multi-Platform Multi-Sensor System Calibration (2012–2016).

Chenglu Wen (M'14) received the Ph.D. degree in mechanical engineering from China Agricultural University, Beijing, China, in 2009.

She is an Associate Professor with Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Information Science and Engineering, Xiamen University, Xiamen, China. She is the author or coauthor of more than 30 research papers published in refereed journals and proceedings. Her research interests include machine vision, machine learning, mobile laser scanning systems, and point cloud data processing.

Dr. Wen serves as the Secretary of the ISPRS WG I/3 on Multi-Platform Multi-Sensor System Calibration (2012–2016).