Automated Extraction of Urban Road Facilities Using Mobile Laser Scanning Data

Article *in* IEEE Transactions on Intelligent Transportation Systems · August 2015

DOI: 10.1109/TITS.2015.2399492

4 authors:

Yongtao Yu
Huaiyin Institute of Technology
41 PUBLICATIONS **370** CITATIONS
SEE PROFILE

Jonathan Li
University of Waterloo
252 PUBLICATIONS **3,240** CITATIONS
SEE PROFILE

Haiyan Guan
Nanjing University of Information Science & …
52 PUBLICATIONS **428** CITATIONS
SEE PROFILE

Cheng Wang
Xiamen University
157 PUBLICATIONS **894** CITATIONS
SEE PROFILE

Automated Extraction of Urban Road Facilities Using Mobile Laser Scanning Data

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

***Abstract*—This paper proposes a novel, automated algorithm for rapidly extracting urban road facilities, including street light poles, traffic signposts, and bus stations, for transportation-related applications. A detailed description and implementation of the proposed algorithm is provided using mobile laser scanning data collected by a state-of-the-art RIEGL VMX-450 system. First, to reduce the quantity of data to be handled, a fast voxel-based upward growing method is developed to remove ground points. Then, off-ground points are clustered and segmented into individual objects via Euclidean distance clustering and voxel-based normalized cut segmentation, respectively. Finally, a 3-D object matching framework, benefiting from a locally affine-invariant geometric constraint, is developed to achieve the extraction of 3-D objects. Quantitative evaluations show that the proposed algorithm attains an average completeness, correctness, quality, and F_1-measure of 0.949, 0.971, 0.922, and 0.960, respectively, in extracting 3-D light poles, traffic signposts, and bus stations. Comparative studies demonstrate the efficiency and feasibility of the proposed algorithm for automated and rapid extraction of urban road facilities.**

***Index Terms*—Bus station, light pole, mobile laser scanning (MLS), road feature inventory, traffic safety, traffic signpost.**

I. Introduction

RAPID update of the inventory of urban road features, on a regular basis, is of great importance for transportation infrastructure management departments, as well as for intelligent transportation-related applications, including driver assistance and safety warning systems [1], [2] and autonomous driving [3], [4]. Accurate and real-time information regarding current road conditions, traffic flow, and the surrounding environment forms significant inputs to the Intelligent Transportation Systems. The absence, or lack of visibility of necessary traffic signals, is often the cause of terrible traffic accidents. Thus, effective advance detecting and monitoring of potential disasters on the road can reduce casualties and improve traffic safety. According to manuals in different countries, documentation of urban road feature inventory includes road geometries (e.g., longitudinal and transverse slopes, road curvatures, lane width, number, and direction of travel), road surface features (e.g., road markings, manholes, sewer wells, and cracks), and roadside infrastructure (e.g., light poles, traffic signposts, power lines, and bus stations). Not only can the vectorized data be considered by transportation agencies to maintain, repair, and reconstruct current road signals, but they also provide auxiliary information for intelligent vehicle applications to make decisions and improve driving safety. Thus, effective, automated extraction of urban road facilities, such as street light poles, traffic signposts, and bus stations, can assist in rapid update of urban road feature inventories.

Traditionally, urban road feature inventory updates are based on field work, where workers from transportation agencies conduct on-site inspections and measurements. This approach for updating large-scale road networks is labor intensive, time consuming, and costly. In addition, it is greatly dangerous for the workers to work on highways or in tunnels. As a revolutionary progress, mobile mapping systems, mounted with digital cameras or videos [5], [6], have been developed and extensively used in a variety of transportation-related activities. However, such systems, using optical imaging, suffer greatly from distortions, environmental illumination conditions, occlusions caused by pedestrians and vehicles, shadows cast by buildings and trees, and lack of accurate georeferencing. In addition, mapping missions are only limited to the daytime.

Laser scanning systems, benefiting from the development of laser and positioning technologies, are now manufactured and widely used in the fields of transportation, forestry, heritage conservation, smart city construction, etc. A mobile laser scanning (MLS) system [7], integrated with laser scanners and global navigation satellite systems (GNSS), uses near-infrared spectra to measure the surface topologies of visible targets. The geospatial information of the measured targets is recorded by a set of 3-D points with real-world position and orientation information (namely, point cloud); the backscattered energy is recorded in a form of intensity for reflecting the targets' surface and material properties. Compared with traditional optical imaging-based systems, MLS systems can acquire real-world coordinates of a large area (in the form of highly dense and

Manuscript received May 9, 2014; revised July 27, 2014, October 29, 2014, and January 22, 2015; accepted February 1, 2015. Date of publication February 24, 2015; date of current version July 31, 2015. This work was supported in part by the general grant from the Natural Science Foundation of China under Grant 41471379 and in part by the discovery Grant from the Natural Sciences and Engineering Research Council of Canada under Grant 111368 awarded to the corresponding author, Jonathan Li (junli@uwaterloo.ca). The Associate Editor for this paper was T. Nakatsuji.

Y. Yu is with the Fujian Key Laboratory of Sensing and Computing for Smart Cities, Xiamen University, Xiamen 361005, China (e-mail: allennessy.yu@gmail.com).

J. Li is with the Key Laboratory of Underwater Acoustic Communication and Marine Information Technology (MOE), 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 is with the School of Information Science and Engineering, Xiamen University, Xiamen 361005, China (e-mail: cwang@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.2015.2399492

accurate 3-D point clouds) in a short time period. MLS systems are immune to the impact of environmental illumination conditions, and can function day and night. Moreover, MLS systems provide direct views of road surfaces and roadside features. Therefore, MLS systems are a promising and feasible alternative to assist in rapidly updating urban road feature inventories.

In this paper, we develop an automated algorithm for extracting urban road facilities using a state-of-the-art RIEGL VMX-450 MLS system. This study mainly focuses on the extraction of street light poles, traffic signposts, and bus stations. Street lamps provide pedestrians and vehicles illumination at night for clear visibility of the road environment and safe driving. Traffic signposts, an important component in transportation, function to control and regulate traffic activities, warn drivers of potential dangers, and exhibit correct routes. Bus stations, a specific segment of a city's infrastructure, play an important role in providing passengers with bus stops and buses and taxis with valid parking spots. However, the absence, or lack of visibility of these fixtures, particularly street light poles and traffic signposts, not only brings inconvenience to transportation-related activities, but also leads to catastrophic traffic accidents. Therefore, accurate and cost-effective methods are urgently in demand for rapid and automated extraction of the aforementioned fixtures.

The contributions of this paper are as follows: 1) a voxel-based upward growing method for removing ground points; 2) a voxel-based normalized cut segmentation method for segmenting the clusters containing multiple objects; and 3) a 3-D object matching framework for extracting objects of interest.

II. Related Work

MLS systems have enriched the corpus of 3-D data available for a variety of transportation-related applications [8]. Highly dense, accurate 3-D point clouds, collected by MLS systems, have become a leading source for highway mapping, tunnel deformation monitoring, bridge distress assessment, etc.

In the following sections, we present a detailed review of existing methods for point cloud segmentation, 3-D object matching, ground removal, and extraction of light poles, traffic signposts, and bus stations using MLS data.

A. Studies on Point Cloud Segmentation

Point cloud segmentation has been extensively studied in the literature. Typically, the existing methods are classified into the following two categories: indoor scene segmentation and natural scene segmentation.

Indoor scene segmentation. In [9], a graphical model that captures various features and contextual relations was proposed to assist in segmenting indoor scenes. With object affordance (e.g., pushable and liftable), a hierarchical technique was developed in [10] to segment indoor scenes for robot manipulation. First, geometric features were extracted from point cloud segments. Then, a classifier was built to predict associated object affordances. Finally, with the classifier, object segmentation results were enhanced and manipulation uncertainty was reduced through iterative clustering and entropy minimization. In [11], a discriminative model that integrates geometrical information, pairwise and higher order components was presented for segmenting indoor scenes. To deal with erroneous sensor inputs, an improved geometry estimation scheme was proposed to improve segmentation accuracy. A feature descriptor, combined with fast point feature histograms (FPFHs) and discriminative graphical models, was used in [12] to depict different geometric surface primitives for the segmentation of indoor scene point clouds. By defining the classes of geometric surfaces and using the contextual information from conditional random fields, this method achieved promising results in indoor scene segmentation.

Natural scene segmentation. Compared with indoor scenes, natural scenes contain more complex and irregular-shaped objects, thereby resulting in more challenges to segment natural point cloud scenes. In [13], an efficient two-step segmentation method was proposed for large-scale point clouds. First, a scan-line-based ground segmentation algorithm was designed to filter ground points. Then, a self-adaptive Euclidean clustering algorithm was used to further separate off-ground points into different objects. A shape-based segmentation method for segmenting urban scene point clouds was developed in [14]. First, this method calculated the geometric features of each point and classified the points based on their geometric features using support vector machines. Then, a set of rules was defined to segment the classified points, and a similarity criterion was proposed to overcome over-segmentation. Finally, based on topological connectivity, segmentation output was merged into meaningful geometrical abstractions. To deal with noisy, varying density point clouds, a multiscale and hierarchical framework [15] was presented to segment urban scene point clouds. First, the point clouds were resampled into different scales. Then, each scale's resampled point set was aggregated into several hierarchical point clusters. Finally, based on a latent Dirichlet allocation model, the hierarchical point clusters were classified into semantic regions by using an AdaBoost classifier.

B. Studies on 3-D Object Matching

Object matching in 3-D patterns has been successfully studied in the literature [16]. Based on the different representations of geometric shapes (e.g., triangle meshes, contours, and point sets), object matching methods are divided into the following three broad categories [17]: feature-based method [18], graph-based method [19], and geometry-based method [20]. A variety of approaches, such as bilateral maps [21], pairwise harmonics [22], curve analyses [23], and sparse models [24], were proposed to match objects represented by meshes. Object matching methods have also been used in object retrieval, recognition, and alignment from 3-D scenes. By using a 3-D deformable matching method based on the Markov random field (MRF) deformation model [25], skeletons of complex articulated objects were extracted from 3-D point cloud sequences. The extracted skeletons were then applied to motion capture. Informative, discriminative feature descriptors play a fundamental role in deformable shape analysis. Recently, spectral descriptors [26] were developed for deformable shape matching purposes.

C. Studies on Ground Removal

Due to the characteristics of MLS systems in direct views of the ground and high laser pulse repetition rate (PRR), ground points occupy a great portion of the resultant point clouds. Such a large number of highly dense ground points not only enlarge the searching regions for extracting off-ground objects, but also slow the processing speed. Therefore, some research has been conducted to develop rapid, effective methods to segment or remove ground points.

In [27], a curb-based approach was developed to segment road surface points. This approach operated rapidly and achieved good performance in segmenting road surface points. However, this approach was limited to only road surface points rather than the entire ground points, and it required the assistance of trajectory data. In [28], random sampling consensus (RANSAC) was first used to fit ground planes on a given point cloud. Then, the points close to the ground planes were labeled as the ground. Similarly, a plane fitting approach was proposed in [29] for removing ground points based on the assumption that ground points construct good plane structures. However, when handling scenes with strong ground fluctuations, such plane fitting-based methods encountered problems. A maximum local slop filter was developed in [30] to filter ground points. This filter identified ground points by comparing the local slopes between a point and its neighbors. In [31], an iterative progressive morphological filter was proposed to remove nonground points. By gradually increasing the window size and using elevation difference thresholds, this filter removed different-sized nonground objects while preserving ground points.

D. Studies on Light Pole Extraction

Existing methods for extracting street light poles are mostly based on pole-like features. However, when dealing with light poles attached with traffic signs or advertising boards or light poles overlapped with or hidden in trees, such methods encounter problems. In [32], a principal component analysis (PCA) method was proposed to extract light poles by considering both shape and context features. Shape features of each light pole were computed using height, number of segments, and structure types; whereas context features were obtained using the surrounding distributions of light poles. Similarly, a covariance-based procedure, based on eigenvalue analysis, was proposed in [33] to detect light poles. Considering prior knowledge of the shape, height, and size of light poles, a percentile-based algorithm was introduced in [34]. This algorithm first divided an object segment into quartiles. Then, the third quartile was selected and partitioned into horizontal slices for further processing. Finally, light poles were recognized based on the detection of vertical pole-like structures from the third quartile.

Rather than dealing with highly dense point clouds in 3-D space, some researchers projected 3-D point clouds into 2-D space. In [35], all clustered objects were first projected onto the horizontal plane. Then, the distribution of each object was captured through covariance matrix decomposition. Finally, light poles were simply extracted based on eigenvalue analysis. In [36], a density of projected points (DoPP) algorithm was adopted for extracting light poles. First, a point cloud was partitioned into voxels on the XY plane. Then, the maximum elevation of each voxel was computed, and an elevation threshold was determined accordingly. Finally, by classifying the point cloud into ground, low ground, and high ground points, light poles were extracted on the classified points via elevation thresholding.

To achieve better visibility of the entire surveyed scene, line scan mode is designed for MLS systems. Consequently, a scan line-based method was presented in [37] for analyzing pole-like structures within segmented profiles. Light poles were successfully extracted through an integration of scan line segmentation, point clustering, cluster merging, and cluster classification. In [38], vertical scan lines on profiles were used to detect point groups that formed independent poles. Similarly, vertical scan lines were also considered in [39] for extracting light poles. However, the extraction was limited to light poles in front of buildings and on the ground.

E. Studies on Traffic Signpost Extraction

Traffic signpost extraction from MLS point clouds has attracted extensive attention in the literature. In [35], based on the knowledge that traffic signposts are usually located close to road boundaries in urban areas, redundant points belonging to road surfaces were first removed by analyzing the scan lines of cross sections. Then, the remaining points were clustered into different spatial objects based on point distances. Finally, traffic signposts were extracted by finding linear features from the projection of each clustered spatial object in the horizontal plane. A smoothing and PCA algorithm was developed in [40] for extracting traffic signposts based on Laplacian smoothing and PCA. Laplacian smoothing, using k-nearest neighbors graph, aimed to smooth each segment to suppress measurement noise and point distribution bias. Then, PCA was applied to the smoothed segments to infer pole-like objects. Hough forest was proposed in [41] for learning mappings from 3-D local patches to object centroids. This method obtained promising results for extracting traffic signposts from MLS data.

Traffic signposts exhibit high retro-reflectivity in MLS point clouds; thus, intensity information provides an important clue for separating traffic signposts from other objects. In [42], a point cloud was first filtered based on the distance to the sensor, sensor angle, and intensity. Then, the filtered points were clustered and divided into a grid structure. Finally, by using the RANSAC algorithm, traffic signposts were extracted by fitting a plane to each cluster. Similarly, traffic signposts in [34] were fitted with their convex hulls, minimum bounding boxes, and minimum bounding circles. In [43], a template-driven method was developed to extract traffic signposts based on their properties of highly reflective planes aligning perpendicularly to the direction of travel and symmetrical shapes.

F. Studies on Bus Station Extraction

Bus station localization is an important issue in public transportation. Information regarding the exact location of bus stations provides useful clues, not only for buses to locate correct

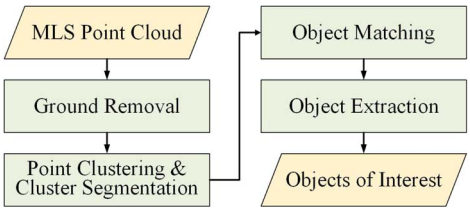


Fig. 1. Flowchart of the proposed algorithm.

parking spots, but also for passengers to quickly find the nearest bus stops. In addition, overall bus station location information within a region can be used by transportation administrators to assess the accessibility of a bus service and improve the coverage of bus routes. However, there are few studies focusing on the extraction and localization of bus stations using MLS data. Therefore, exploiting effective means and methods for bus station extraction from MLS data is an interesting and meaningful research topic.

III. Method

Fig. 1 shows the detailed workflow of our proposed algorithm. The input MLS point clouds are first preprocessed to remove ground points using a voxel-based upward growing method. Then, the off-ground points are clustered and segmented into individual objects via Euclidean distance clustering and voxel-based normalized cut segmentation, respectively. Finally, the objects of interest are extracted based on a 3-D object matching framework.

A. Ground Removal

To obtain good coverage of the surveyed scene, MLS systems mounted with two laser scanners are basically configured with a "Butterfly" (or "X") pattern. While the mapping vehicle is being driven, these two laser scanners rotate to emit laser beams. Such a scan mode results in a grid-like pattern in the scanned point clouds. Due to the laser scanners' properties in direct ground views and high measurement rates, ground points consume a great portion of the collected point clouds. Because the objects of interest to be extracted belong to off-ground objects, effectively separating ground and off-ground points in the point cloud is helpful to reduce the searching regions. Therefore, to reduce the complexities of both time and space, we develop a novel voxel-based upward growing method that rapidly and effectively filters ground points from the entire point cloud.

Considering the fluctuations of the ground and the efficiency of our proposed ground removal method, we first vertically divide a raw point cloud into a set of data blocks with a size of w_b in the XY plane. These data blocks are processed separately to remove ground points. Then, we organize each of the data blocks into an octree structure with a voxel size of w_v [see Fig. 2(a)]. As shown in Fig. 2(b), by using the octree partition structure, a voxel is connected with 26 neighbors in 3-D space. Based on such an octree partition structure, our voxel-based upward growing method is carried out as follows.

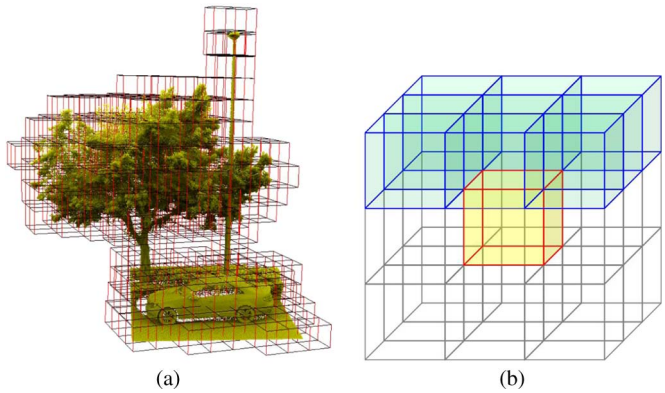


Fig. 2. (a) Octree partition structure, and (b) upward growing scheme.

Fig. 3. (a) Raw point cloud, and (b) off-ground points after ground removal.

For each voxel v in a data block, it grows upward to its nine neighbors, who are connected with this voxel and located above this voxel [see Fig. 2(b)]. Then, the same growing scheme is applied to the neighbors to continuously grow upward. The growing process stops when no more voxels can be reached according to the upward growing scheme. Finally, the voxel v_h that has the highest elevation within the grown region is ascertained. If the elevation of v_h is smaller than a predefined ground threshold h_g, voxel v is regarded as ground; then, all the points in voxel v are removed. However, if the elevation of v_h is larger than h_g, voxel v is treated as a part of an off-ground object; then, all the points in voxel v are retained. The voxel-based upward growing method for ground removal has the following properties: 1) suitable for handling large scenes with strong ground fluctuations; and 2) removes ground points rapidly and effectively. Fig. 3 shows a visual example of the point cloud before and after ground removal.

B. Off-Ground Point Clustering and Segmentation

After ground points are filtered out, the off-ground points remain isolated and unorganized. The points belonging to a specific object must be further grouped together before identifying the objects of interest. In this paper, we introduce a Euclidean distance clustering approach, which groups discrete points based on their Euclidean distances to their neighbors. Particularly, an unlabeled point is grouped into a specific cluster if and only if its shortest Euclidean distance to the points within this cluster lies below a clustering threshold d_c. Otherwise, a new cluster is created to contain this point.

The Euclidean distance clustering approach starts at an arbitrary unlabeled point and expands radially outward to its neighbors within the spherical region indicated by a radius size d_c. The clustering procedure continues outward from its neighbors

Fig. 4. (a) Clustering result of off-ground points, and (b) cluster filtering result using prior knowledge.

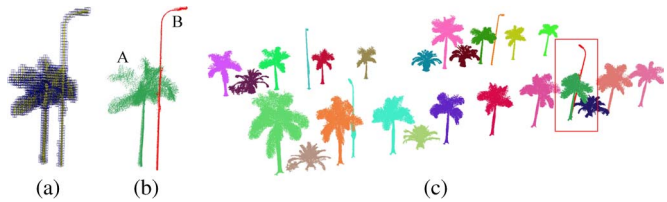


Fig. 5. (a) Octree partition strategy, (b) illustration of bipartition, and (c) voxel-based normalized cut segmentation result.

to their corresponding neighbors. Such a recursive clustering procedure stops when no more points can be contained in this cluster. Then, the remaining unlabeled points are further grouped into specific clusters by using the same clustering scheme. Fig. 4(a) presents a clustering result of the off-ground points. In Fig. 4, different colors represent different clusters. Specifically, to reduce the computational complexity, based on prior geometric knowledge (e.g., height and size) of the objects of interest to be extracted, the small clusters that are unlikely targets are further removed. Fig. 4(b) shows the filtered clusters using prior geometric knowledge.

However, as cluster A in Fig. 4(b) shows, by using the Euclidean distance clustering approach, the objects (a tree and a light pole) that are adjacent or overlap cannot be effectively separated. To further segment such clusters containing more than one object, we develop a novel voxel-based normalized cut segmentation method. Normalized cut segmentation [44] has proved to be an accurate, effective method for segmenting 2-D images. In this paper, we modify this method to achieve the segmentation of 3-D objects in 3-D point clouds.

First, the clusters containing more than one object are divided into a voxel structure with a voxel spacing w_s using the octree partition strategy [see Fig. 5(a)]. Then, all the voxels are constructed into a complete weighted graph $\boldsymbol{G} = \{\boldsymbol{V}, \boldsymbol{E}\}$, where the vertices $\boldsymbol{V}$ are represented by the voxels, and edges $\boldsymbol{E}$ are connected between each pair of voxels. The weight on the edge is used for measuring the similarity between a pair of voxels connected by the edge. To assign a meaningful, distinctive weight w_{ij} to the edge connecting a pair of voxels $(i, j) \in \boldsymbol{V}^2$, we compute the weight from the features associated with the voxels as follows:

$$w_{ij} = \begin{cases} \exp\left(-\frac{\left\|p_i^H - p_j^H\right\|_2^2}{\sigma_H^2}\right) \cdot \exp\left(-\frac{|p_i^Z - p_j^Z|^2}{\sigma_Z^2}\right) \\ \qquad\qquad \text{if } \left\|p_i^H - p_j^H\right\|_2 \leqslant d_H \\ 0, \qquad\qquad \text{otherwise,} \end{cases} \tag{1}$$

where $p_i = (x_i, y_i, z_i)$ and $p_j = (x_j, y_j, z_j)$ are the centroids of voxels i and j, respectively. $p_i^H = (x_i, y_i)$ and $p_j^H = (x_j, y_j)$ are the coordinates of the centroids on the XY plane; $p_i^Z = z_i$ and $p_j^Z = z_j$ are the z coordinates of the centroids. σ_H and σ_Z are the standard deviations of the horizontal and vertical distributions, respectively. d_H is a distance threshold for determining the maximal valid distance between two voxels in the horizontal plane. The centroid of voxel i is defined as

$$p_i = \frac{1}{N_i} \sum_{m=1}^{N_i} p_m^i \tag{2}$$

where N_i denotes the number of points within voxel i, and p_m^i $(m = 1, 2, \ldots, N_i)$ is a point within voxel i. By such a definition, the similarity between two voxels is measured by considering their distance in the horizontal plane and their relative horizontal and vertical distributions.

As shown in Fig. 5(b), normalized cut segmentation aims to partition the complete weighted graph $\boldsymbol{G}$ into two disjoint voxel groups $\boldsymbol{A}$ and $\boldsymbol{B}$ by maximizing the similarity within each voxel group and maximizing the dissimilarity between two voxel groups. The corresponding cost function [44] is expressed as

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)} \tag{3}$$

where $cut(\boldsymbol{A}, \boldsymbol{B})$ denotes the sum of the weights on the edges connecting voxel groups $\boldsymbol{A}$ and $\boldsymbol{B}$; $assoc(\boldsymbol{A}, \boldsymbol{V})$ denotes the sum of the weights on the edges falling in voxel group $\boldsymbol{A}$; and $assoc(\boldsymbol{B}, \boldsymbol{V})$ denotes the sum of the weights on the edges falling in voxel group $\boldsymbol{B}$. The above terms are defined as follows:

$$cut(A, B) = \sum_{i \in A, j \in B} w_{ij} \tag{4}$$

$$assoc(A, V) = \sum_{i \in A, j \in V} w_{ij} \tag{5}$$

$$assoc(B, V) = \sum_{i \in B, j \in V} w_{ij}. \tag{6}$$

The minimization of $Ncut(\boldsymbol{A}, \boldsymbol{B})$ is obtained by solving the corresponding generalized eigenvalue problem [44]

$$(\mathbf{D} - \mathbf{W})y = \lambda \mathbf{D} y \tag{7}$$

where $W(i, j) = w_{ij}$, and $\boldsymbol{D}$ is a diagonal matrix, whose ith row records the sum of the weights on the edges associated with voxel i

$$D(i, j) = \begin{cases} \sum\limits_{m \in V} w_{im}, & \text{if } i = j \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

After carrying out eigenvalue decomposition on (7), a set of eigenvalues and their associated eigenvectors are obtained. Finally, according to the normalized cut segmentation scheme, we partition the cluster into two segments by applying a threshold to the eigenvector associated with the second smallest eigenvalue [44]. The segmentation result is shown in Fig. 5(c).

C. Three-Dimensional Object Matching Framework

To handle the conditions of affine transformations, occlusions, incompleteness, and scales existing in the objects from the same category in 3-D MLS point clouds, we propose a novel 3-D object matching framework to support object extraction. The problem of object matching in MLS point clouds is depicted as matching a set of template feature points, representing an object, to a set of scene feature points, representing a scene containing an instance of that object. Denote n_T and n_S ($n_T \leq n_S$) as the number of template and scene feature points, respectively. Let $\mathbf{T} \in R^{n_T \times 3}$ and $\mathbf{S} \in R^{n_S \times 3}$ be the matrices recording the 3-D coordinates of the template and scene feature points, respectively. The objective of object matching is to ascertain an optimal matching function $f(\cdot)$ that 1) matches every template feature point $p_i = (x_i, y_i, z_i) \in R^3, i = 1, 2, \ldots, n_T$ to a scene feature point $q_j = (x_j, y_j, z_j) \in R^3$, $j = 1, 2, \ldots, n_S$, and 2) minimizes the following overall objective function containing both feature and geometric matching costs:

$$\min_f \sum_{i=1}^{n_T} \left[c\left(p_i, f(p_i)\right) + \lambda \cdot g\left(p_i, N(p_i); f(p_i), f\left(N(p_i)\right)\right) \right] \tag{9}$$

where $c(p, q)$ denotes the feature matching cost between template feature point p and scene feature point q; $N(p_i)$ denotes the neighbors of point p_i; $g(\cdot)$ is a geometric matching cost function, which measures the dissimilarity between two sets of points $\{p_i, N(p_i)\}$ and $\{f(p_i), f(N(p_i))\}$; λ balances the relative weight between the feature and the geometric matching costs.

1. Feature Matching Cost: Because both the template and the scene are represented by discrete and irregularly distributed point sets, the feature descriptors selected for modeling the local features of the feature points must possess the properties of robustness to noise and high distinctiveness to different feature points. FPFHs [45] have proved to be a promising descriptor to rapidly and saliently depict feature points in 3-D point clouds. Generally, the FPFH descriptor generates a 16-D histogram descriptor for each feature point. Denote $\boldsymbol{H}_p$ as the histogram descriptor for feature point p. Then, the cost of matching a template feature point p to a scene feature point q is defined using the χ^2 distance [46] as follows:

$$c(p, q) = \sum_{k=1}^{16} \frac{\left[H_p(k) - H_q(k)\right]^2}{H_p(k) + H_q(k)}. \tag{10}$$

The feature matching costs between each pair of template and scene feature points are stored in a feature matching cost matrix $\mathbf{C} \in R^{n_T \times n_S}$. The element $C_{ij} = c(p_i, q_j)$, $i = 1, 2, \ldots, n_T$, $j = 1, 2, \ldots, n_S$ stores the cost of matching the ith template feature point p_i to the jth scene feature point q_j.

The matching function $f(\cdot)$ is usually modeled as a binary variable matrix [47] $\mathbf{X} \in \{0, 1\}^{n_T \times n_S}$

$$X_{ij} = 1 \Leftrightarrow f(p_i) = q_j, i = 1, 2, \ldots, n_T, j = 1, 2, \ldots, n_S. \tag{11}$$

$X_{ij} = 1$ indicates that the ith template feature point p_i is matched to the jth scene feature point q_j; $X_{ij} = 0$ means that there is no matching between p_i and q_j. Because each template feature point is matched to exactly one scene feature point, each row of X contains exactly one element with a value of one.

By integrating the feature matching cost matrix $\mathbf{C}$ and the binary variable matrix $\mathbf{X}$, the feature matching cost term in (9) is expressed as

$$\sum_{i=1}^{n_T} c\left(p_i, f(p_i)\right) = tr\left(\mathbf{C}\mathbf{X}^T\right) = \sum_{i=1}^{n_T} \sum_{j=1}^{n_S} C_{ij} X_{ij}. \tag{12}$$

Because each row of $\mathbf{X}$ contains exactly one element with a value of one, only one matching cost for each template feature point is aggregated into the feature matching cost term.

For each row of $\mathbf{X}$, the column index of the element with a value of one in this row specifies the index of the scene feature point that is matched to the corresponding template feature point. Rewrite $\mathbf{X}$ into a row-vector representation as follows:

$$\mathbf{X}^T = \left[\mathbf{X}_1^T, \mathbf{X}_2^T, \ldots \mathbf{X}_{n_T}^T\right] \tag{13}$$

where $\mathbf{X}_i$, $i = 1, 2, \ldots, n_T$ denotes the ith row of $\mathbf{X}$. Then, $\mathbf{X}_i\mathbf{S}$ computes the 3-D coordinates of the matched scene feature point for template feature point p_i. Combining all rows of $\mathbf{X}$, $\mathbf{XS}$ represents the 3-D coordinates of all the matched scene feature points in the same order as the template feature points.

2. Geometric Matching Cost: Considering the affine transformations, occlusions, incompleteness, and scales among the objects from the same category, we propose a locally affine-invariant geometric constraint to model the geometric matching cost function $g(\cdot)$ in (9). We assume that each template feature point p_i is exactly represented by an affine combination of its neighbors as follows:

$$p_i = \sum_{p_j \in N(p_i)} A_{ij} p_j \tag{14}$$

where $\mathbf{A}$ is an $n_T \times n_T$ weight matrix recording the affine combination coefficients for all template feature points. Denote $\mathbf{A}_i$ as the ith row of $\mathbf{A}$. Then, $\boldsymbol{A}_i$ models the local geometric distribution around p_i. The following two constraints are placed on $\mathbf{A}$: 1) $A_{ij} = 0$ if $p_j \notin N(p_i)$; and 2) each row of $\mathbf{A}$ must add up to one. The first constraint reflects the local geometric properties of each template feature point; the second constraint makes the representation invariant to global transformations. These two constraints guarantee that a template feature point is always exactly represented by an affine combination of its neighbors.

Ideally, by using the affine combination, the representation error for each template feature point p_i is always zero under the L_1-norm

$$\left\| p_i - \sum_{p_j \in N(p_i)} A_{ij} p_j \right\|_1 = 0, \qquad i = 1, 2, \ldots, n_T. \tag{15}$$

Summing up the representation errors from all template feature points results in the following expression:

$$\|\mathbf{T} - \mathbf{A}\mathbf{T}\|_1 = \|(\mathbf{I} - \mathbf{A})\mathbf{T}\|_1 = 0 \tag{16}$$

where $\mathbf{I}$ is an identity matrix, and $\|\cdot\|_1$ denotes the summation of the absolute values of all the elements in a matrix.

Because $\mathbf{XS}$ represents the matched scene feature points in the same order as the template feature points, replacing $\mathbf{T}$ with

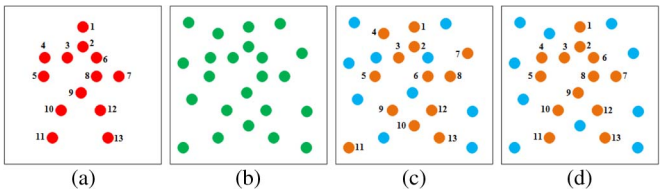


Fig. 6. Illustration of the LPSA algorithm: (a) template feature points, (b) scene feature points, (c) initial matching with linear programming, and (d) final matching refined by simulated annealing.

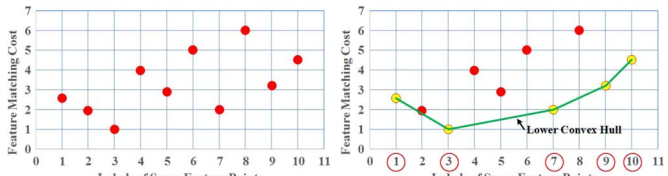


Fig. 7. Illustration of lower convex hull. (a) Two-dimensional point cloud formulation of scene feature points, and (b) lower convex hull and the matching candidates.

$\mathbf{XS}$ in (16) results in the following geometric matching cost term in (9):

$$\sum_{i=1}^{n_T} g\left(p_i, N(p_i); f(p_i), f\left(N(p_i)\right)\right) = \|(\mathbf{I}-\mathbf{A})\mathbf{XS}\|_1 . \quad (17)$$

3. Overall Objective Function: By embedding the feature matching cost term in (12) and the geometric matching cost term in (17), the overall objective function for 3-D object matching is defined as follows:

$$\begin{aligned} \underset{\mathbf{X}}{\text{minimize}} \quad & tr(\mathbf{C}\mathbf{X}^T) + \lambda \|(\mathbf{I}-\mathbf{A})\mathbf{XS}\|_1 \\ \text{subject to} \quad & \mathbf{X} \in \{0,1\}^{n_T \times n_S}, \mathbf{X}\mathbf{1}_{n_S} = \mathbf{1}_{n_T}, \mathbf{X}^T\mathbf{1}_{n_T} \leqslant \mathbf{1}_{n_S}. \end{aligned} \quad (18)$$

There are three constraints in (18).

a) $\mathbf{X} \in \{0,1\}^{n_T \times n_S}$ indicates a point-to-point matching pattern, where the matching between a pair of feature points is either "successful" $(X_{ij} = 1)$ or "failed" $(X_{ij} = 0)$.
b) $\mathbf{X}\mathbf{1}_{n_S} = \mathbf{1}_{n_T}$ indicates that each template feature point must be matched to exactly one scene feature point.
c) $\mathbf{X}^T\mathbf{1}_{n_T} \leqslant \mathbf{1}_{n_S}$ indicates a one-to-one matching pattern, where each scene feature point is matched by, at most, one template feature point, i.e., $f(p) = f(q) \Leftrightarrow p = q$.

The overall objective function in (18) has a nonlinear form with integer constraints. Such a formulation results in an NP-hard problem, which cannot be efficiently solved in polynomial time. To obtain an approximately optimal solution to this NP-hard problem, we develop a linear programming and simulated annealing (LPSA) algorithm. The LPSA algorithm starts by solving the following relaxed linear programming problem in a continuous space:

$$\begin{aligned} \underset{\mathbf{X}}{\text{minimize}} \quad & tr(\mathbf{C}\mathbf{X}^T) \\ \text{subject to} \quad & \mathbf{X} \in [0,1]^{n_T \times n_S}, \mathbf{X}\mathbf{1}_{n_S} = \mathbf{1}_{n_T}, \mathbf{X}^T\mathbf{1}_{n_T} \leqslant \mathbf{1}_{n_S}. \end{aligned} \quad (19)$$

This linear programing problem considers only the feature matching cost for obtaining an initial solution. Solving such a linear programming problem leads to a continuous solution of $\mathbf{X}$. To map the continuous solution to a discrete integer solution space, we simply set the element with the maximal value in each row of $\mathbf{X}$ to one and set the others in this row to zero. As shown in Fig. 6(c), some template feature points are mismatched in the initial solution. Then, the initial solution is iteratively refined by a simulated annealing procedure based on (18). This simulated annealing procedure takes into consideration both feature and geometric matching costs. By iteratively adjusting the mismatched feature points based on the geometric constraints, an optimal matching is finally obtained, as shown in Fig. 6(d).

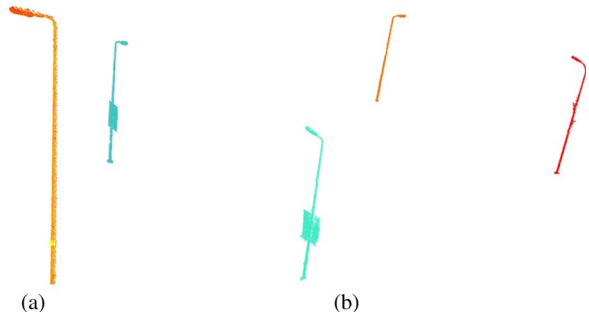


Fig. 8. (a) Light pole prototype, and (b) extracted light poles.

The LPSA algorithm operates efficiently when the numbers of both the template and scene feature points are small. The size of the binary variable matrix $\mathbf{X}$ increases dramatically as the number of feature points increases, resulting in a great computational burden in solving the linear programing problem in (19). To reduce the computational complexity, we adopt a lower convex hull trick [48] to obtain the initial matching solution. For each template feature point p, we regard the scene feature points as a 2-D point cloud with their indices as the x-axis, and the feature matching costs as the y-axis [see Fig. 7(a)]. Then, we compute a lower convex hull with respect to the second dimension [see Fig. 7(b)]. Finally, only the scene feature points on the lower convex hull are considered as the matching candidates for p. Such a lower convex hull trick guarantees that the number of matching candidates associated with each template feature point is dramatically reduced.

D. Three-Dimensional Object Extraction

Based on the 3-D object matching framework in *Section C*, we propose a model-driven method for extracting 3-D objects from the segmented off-ground objects. First, a clean and completely scanned point cloud object prototype [see Fig. 8(a)] is selected for each category of the objects to be extracted. Then, the prototype and the segmented objects are voxelized using the octree partition structure with spacings of v_T and v_S $(v_T > v_S)$, respectively. For each voxel, the point nearest the centroid of this voxel is selected as the feature point. Next, the 3-D object matching framework is applied to the prototype and each segmented object to compute the matching cost between

Fig. 9. Illustration of the RIEGL VMX-450 MLS system and its components.

the feature points on the prototype and the feature points on each segmented object. Finally, the matching costs from all segmented objects are thresholded to obtain the extraction result. Fig. 8(b) shows the extracted light poles from the segmented objects in Fig. 5(c).

IV. Results and Discussion

A. RIEGL VMX-450 System and MLS Point Clouds

In this paper, a state-of-the-art RIEGL VMX-450 MLS system (see Fig. 9) was used to collect 3-D point cloud data along an urban road in Xiamen, China. The mapping vehicle was moving at an average speed of approximately 40–50 km/h while acquiring the point cloud data. Generally, in urban areas, the mapping vehicle can move at a normal city-limited speed (e.g., 40–60 km/h). The RIEGL VMX-450 system smoothly integrates two RIEGL VQ-450 laser scanners, four high-resolution digital cameras, and a set of position and orientation systems, including two GNSS, an inertial measurement unit, and a wheel-mounted distance measurement indicator (see Fig. 9). The two laser scanners are configured with an "X" pattern and rotate to emit laser beams with a maximum measurement rate of 1.1 million measurements per second, a line scan speed of up to 400 scans per second, and a maximum valid range of 800 m. The accuracy and precision of the scanned 3-D point clouds are within 8 and 5 mm, respectively.

The 3-D point cloud data used in this study were acquired on Ring Road South, a typical urban road near the seaside. Four data sets covering the areas of Siming Road South (SRS), Yunding Road South (YRS), International Conference and Exhibition Center (ICEC), and Xianyue Road (XR) were selected for evaluating the 3-D object extraction performance. The SRS data set contains about 581 million points and covers a road length of approximately 4260 m. The YRS data set contains about 421 million points and has a road distance of approximately 2522 m. The ICEC data set contains about 568 million points and has a road segment of approximately 2947 m. The XR data set contains about 273 million points and has a road section of approximately 3693 m.

B. Point Cloud Segmentation

In this paper, we propose a combination of Euclidean distance clustering and voxel-based normalized cut segmentation to group off-ground points into individual objects. Euclidean distance clustering rapidly clusters separated point groups; whereas voxel-based normalized cut segmentation further divides the adjacent or overlapped clusters into separated objects. Here, we conducted a group of experiments to compare the performance of the proposed segmentation method with the two-step segmentation method [13] and the shape-based segmentation method [14]. As shown in Row 1 of Fig. 10, three point cloud scenes were selected for comparative study. After ground removal, the off-ground points were segmented using these three methods. The segmentation results obtained using the two-step method, the shape-based method, and the proposed method are shown in Rows 3, 4, and 5, respectively. On the whole, these three methods all achieved promising segmentation results on the three selected scenes. However, as shown by the boxes labeled #B, #D, #E, and #F, the point cloud clusters containing multiple trees failed to be segmented by using the shape-based method; whereas such clusters were well separated into individual trees by using the two-step and the proposed methods. Moreover, as shown by the box labeled #C, the point cluster containing two overlapped trees could not be segmented with the two-step and the shape-based methods; whereas it was successfully segmented by using the proposed method. In addition, as shown by the box labeled #A, a light pole is hidden in a palm tree and located very closely to the palm tree. The two-step and the proposed methods failed to segment them; however, the light pole and the palm tree were well segmented by using the shape-based method. In conclusion, compared with the other two methods, the shape-based method attains the worst performance. The shape-based method well segments separated clusters and overlapped clusters containing different categories of objects; however, it has problems in segmenting clusters containing closely overlapped objects from the same category. The two-step and the proposed methods well handle the separated and not very closely overlapped clusters; however, they have problems in segmenting the clusters containing very closely overlapped objects either from the same or different categories. Comparatively, the proposed method obtains relatively better performance than the two-step method.

C. Three-Dimensional Object Matching

Here, we evaluate the performance of our proposed 3-D object matching framework to match a template containing a point cloud object to a scene containing an instance of that object. We adopted a voxel-based sampling approach to select a group of feature points from the template and scene point clouds, respectively. First, both the template and scene point clouds were divided into a voxel structure with voxel sizes of v_T and v_S, respectively, based on the octree partition structure. Then, for each of the voxels, the point closest to the centroid of this voxel was chosen as the feature point. Finally, the proposed 3-D object matching framework was applied to the template and scene point clouds to compute correspondences between the template and scene feature points.

Seven data sets containing different categories of objects were selected from the MLS point clouds (see Fig. 11). Table I details the parameter configurations and the object matching performance of these seven data sets. The object matching performance was evaluated by the matching errors represented by

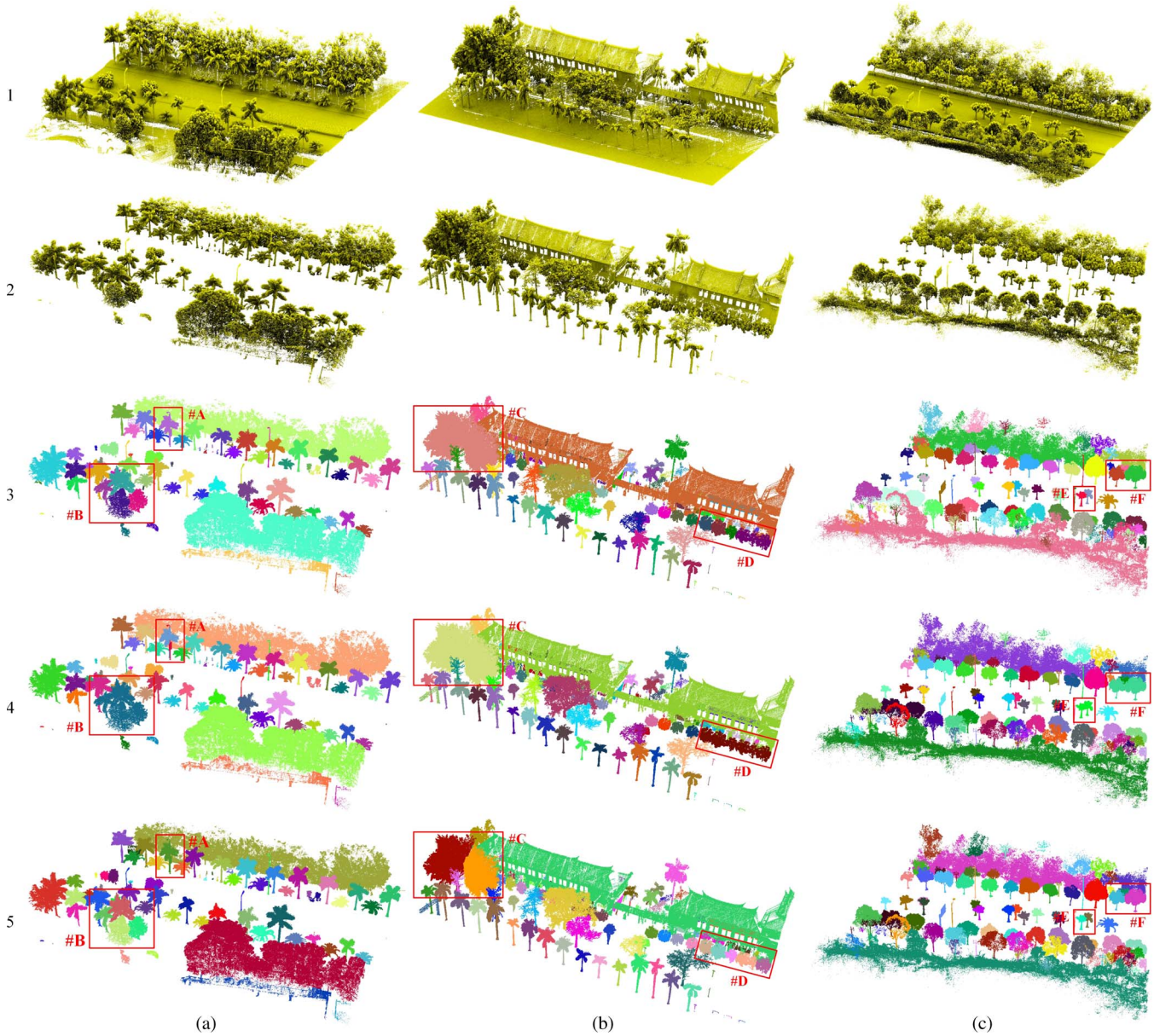


Fig. 10. Point cloud segmentation results on three selected scenes (a), (b), and (c) by using different segmentation methods. Row 1: raw point clouds of the selected scenes. Row 2: off-ground points after ground removal. Row 3: segmentation results obtained using the two-step segmentation method [13]. Row 4: segmentation results obtained using the shape-based method [14]. Row 5: segmentation results obtained using the proposed method.

the percentage of mismatched template feature points. The 3-D object matching results are illustrated in Fig. 11. As reflected by the matching errors in Table I, the proposed 3-D object matching framework obtained promising matching results and achieved very low matching errors on the seven data sets. For all of the seven data sets, the number of scene feature points is much greater than the number of template feature points. Therefore, we used the lower convex hull trick to compute the initial matching for each template feature point. On the whole, the proposed 3-D object matching framework works efficiently in matching a set of template feature points to a relatively large set of scene feature points.

Comparative studies were also conducted to further compare the proposed 3-D object matching framework with several existing methods, including 3-D shape context [49], bilateral map [21], curve analysis [23], spectral descriptor [26], and our proposed framework without local geometric constraints. Fig. 12 shows 12 groups of 3-D point clouds used in the comparative studies. Each group contains a template object and a scene object. The red dots represent the manually sampled salient feature points. The matching performance of different methods was evaluated by the matching errors represented by the percentage of mismatched template feature points (See Fig. 13). Fig. 12 shows the visual examples of the matching results obtained using the proposed 3-D object matching framework. As reflected by the matching performance in Fig. 13, our proposed framework outperforms the other methods and achieves a very low average matching error of 0.79%. Particularly, comparing the average matching errors obtained using our proposed framework with and without local geometric constraints (0.79% versus 51.87%), we conclude that performance is greatly improved when locally affine-invariant geometric

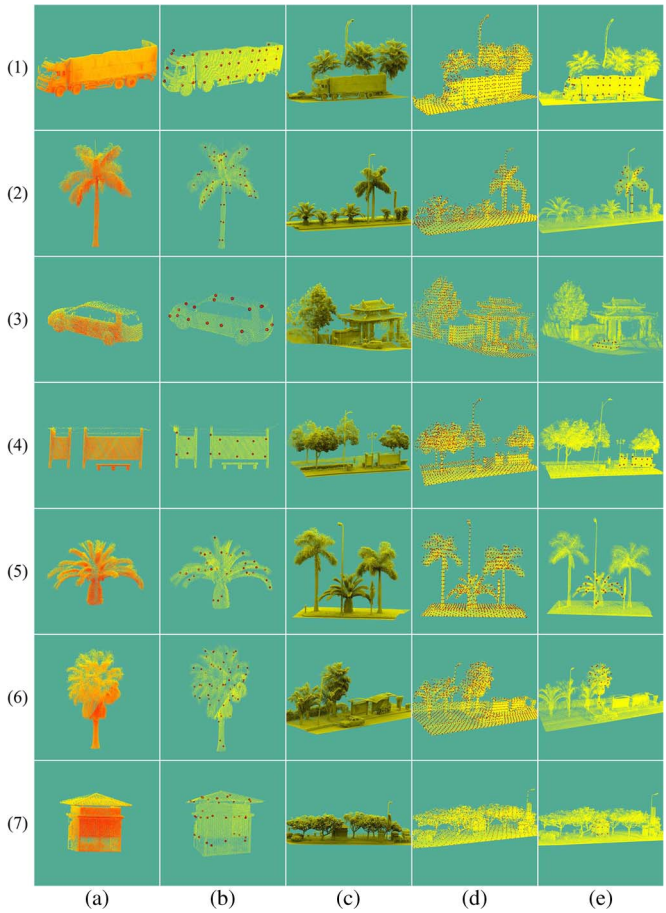


Fig. 11. (a) Template point cloud, (b) template feature points, (c) scene point cloud, (d) scene feature points, and (d) 3-D object matching results.

TABLE I
PARAMETER CONFIGURATIONS AND MATCHING ERRORS

Dataset	v_T (m)	v_S (m)	n_T	n_S	λ	Matching Error (%)
1	1.0	0.3	45	4160	1.0	6.7
2	1.0	0.3	54	3326	1.0	3.7
3	0.5	0.2	17	3348	1.0	0.0
4	1.0	0.3	14	3814	1.0	0.0
5	1.0	0.3	32	2710	1.0	3.1
6	1.0	0.3	53	5810	1.0	3.8
7	1.0	0.3	32	9241	1.0	0.0

constraints are considered. In conclusion, our proposed 3-D object matching framework outperforms the other methods and achieves the best matching performance.

D. Ground Removal

Here, we test the impact of different parameter configurations on the performance of ground removal. Two test scenes were selected from the MLS point clouds. The first test scene (a two-directional-two-lane road) contains 6.8 million points and has a road segment of 48.57 m [see Fig. 14(a)]. The second test scene (a two-directional-four-lane road) contains 12.1 million points and has a road section of 94.48 m [see Fig. 14(b)]. A group of parameter configurations were tested for block size w_b and voxel size w_v. The computing time for removing ground points with each parameter combination is also recorded and detailed in Fig. 15. In this paper, the ground threshold h_g was set at 0.4 m. According to the test results, with a fixed voxel size, the ground removal time increases as the block size increases; whereas with a fixed block size, the ground removal time decreases as the voxel size increases. Therefore,

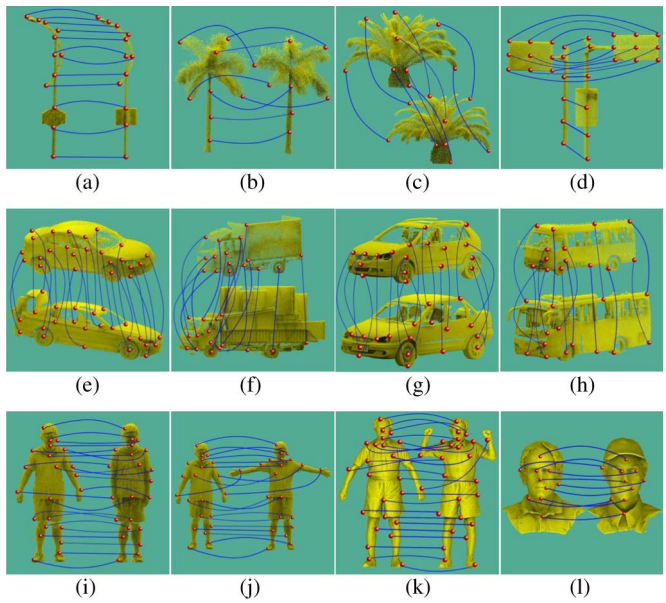


Fig. 12. (a)–(l) Matching results obtained using the proposed framework.

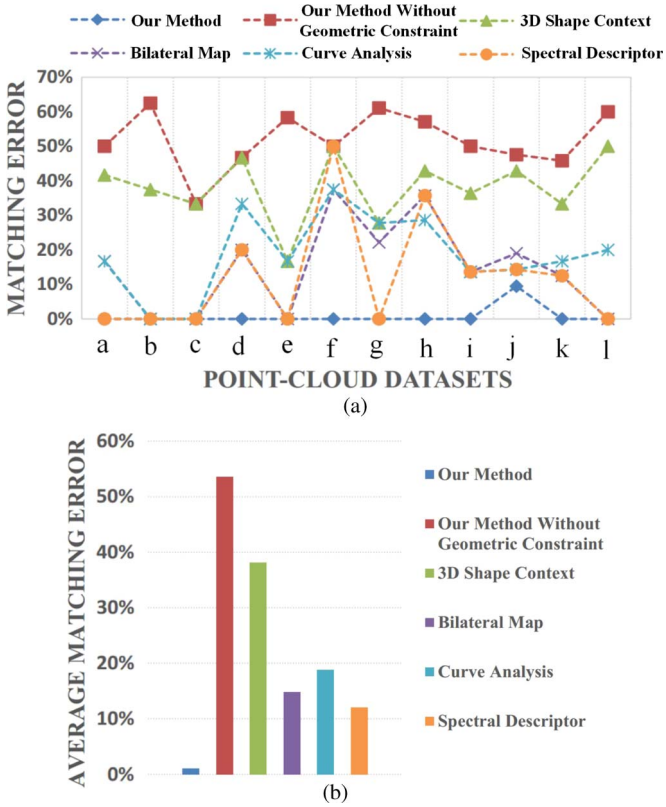


Fig. 13. (a) Matching errors and (b) average matching errors obtained using different methods.

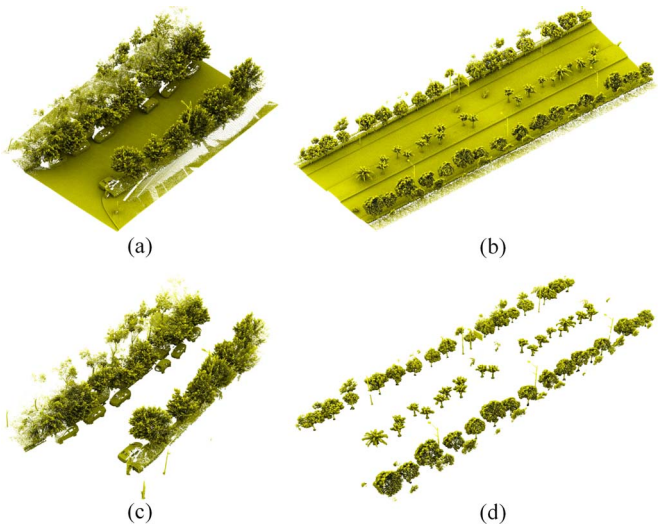


Fig. 14. (a) Test scene one, (b) test scene two, (c) ground removal result of test scene one, and (d) ground removal result of test test scene two.

theoretically, to obtain good performance in rapidly removing ground points, the block size should be selected as small as possible, and the voxel size should be selected as large as possible. However, a large voxel size reduces the accuracy in removing ground points. Therefore, considering both time complexity and accuracy, we chose parameter configurations of $w_b = 3$ m and $w_v = 5$ cm, which provide for good accuracy and low computing time in removing ground points. Fig. 14(c) and (d) show the ground removal results for these two test scenes with $w_b = 3$ m, $w_v = 5$ cm, and $h_g = 0.4$ m.

Comparative studies were also conducted to further evaluate the performance and efficiency between our proposed voxel-based upward growing method and the ground removal method proposed in [28]. As shown in the first row of Fig. 16, four point cloud scenes with different types of ground conditions were selected for comparative study. Fig. 16(a) shows a 1-D flat road with dense vegetation and sidewalks; Fig. 16(b) shows a 2-D flat road with a median separating the travel directions; Fig. 16(c) shows a 4-D sloped road with two low-height and sloped road sections and two overhanging sloped highways; Fig. 16(d) shows a sloped, uneven ground with up-and-down grass lawns. The segmented ground points using the voxel-based upward growing method and the method in [28] are shown in the second and the last rows of Fig. 16, respectively. As shown by the red boxes in Fig. 16, by using the ground removal method in [28], some cars and low-height vegetation were falsely labeled as ground. However, by using the voxel-based upward growing method, such objects were correctly labeled as off-ground objects. Comparatively, the voxel-based upward growing method has the property of effectively removing ground points in both flat and sloped ground conditions, while simultaneously preserving off-ground objects from their bottoms. In addition, computing time was also recorded for efficiency evaluation. By using the voxel-based upward growing method, the ground removal time was 1.71, 3.26, 3.98, and 3.87 s, respectively, on these four selected scenes. However, by using the ground removal method in [28], the computing time

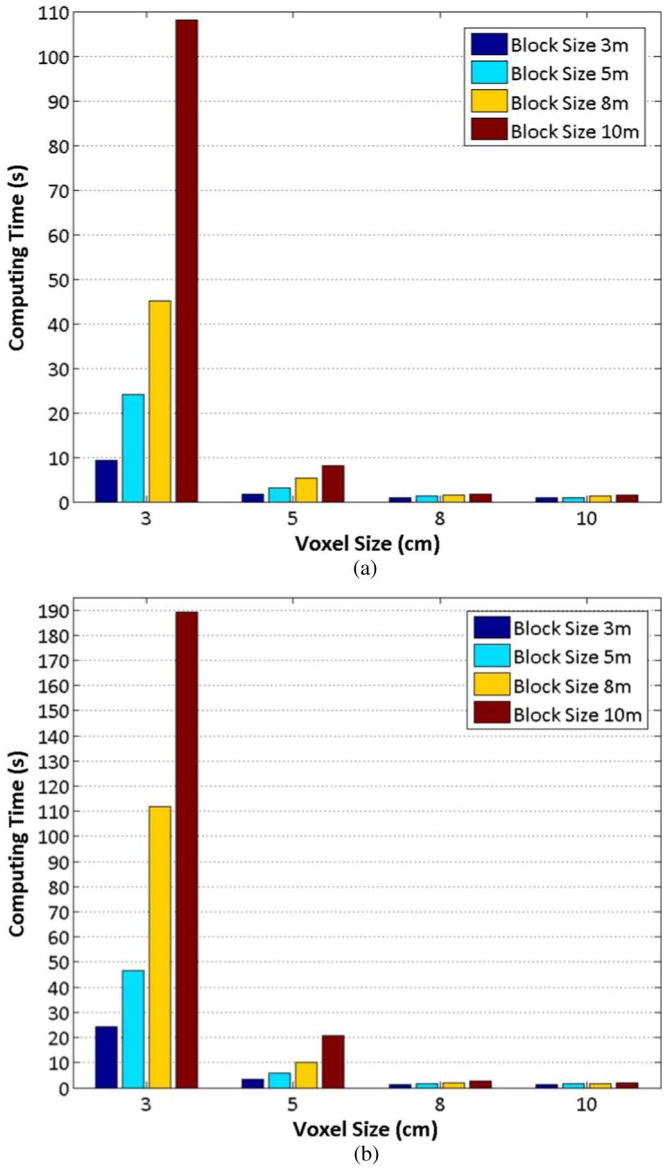


Fig. 15. Ground removal time of different parameter combinations on (a) the first test scene and (b) the second test scene.

was 5.98, 9.66, 9.77, and 10.02 s, respectively, on these four selected scenes. Therefore, the voxel-based upward growing method operates faster and achieves better ground removal results than the method proposed in [28].

E. Three-Dimensional Object Extraction

To evaluate the performance of our proposed algorithm in extracting 3-D light poles, traffic signposts, and bus stations from MLS point clouds, we applied our proposed 3-D object extraction algorithm to the four selected data sets (SRS, YRS, ICEC, and XR data sets) mentioned in *Section A*. The parameter configurations in each processing step are detailed in Table II. To reduce both spatial and computational complexities, these four data sets were first preprocessed to filter out ground points based on the voxel-based upward growing method. Then, the off-ground points were grouped and segmented into clusters

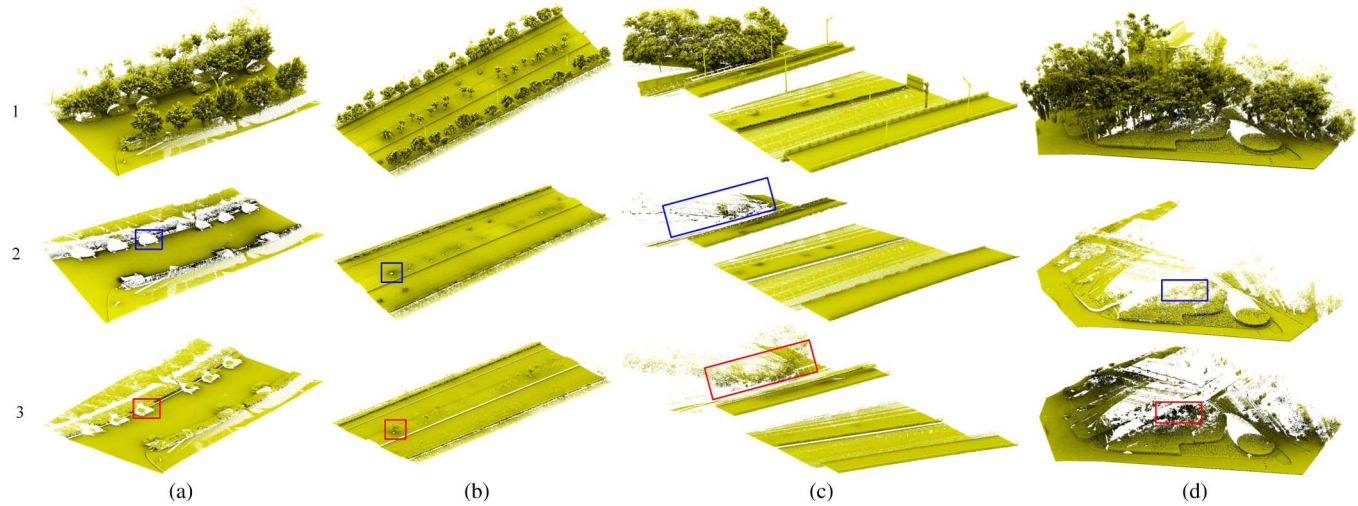


Fig. 16. Ground removal results obtained on four selected scenes (a), (b), (c), and (d) by using different methods. Row 1: raw point clouds of the selected scenes. Row 2: segmented ground points by using the proposed voxel-based upward growing method. Row 3: segmented ground points by using the method in [28].

TABLE II
PARAMETER CONFIGURATIONS IN 3-D OBJECT EXTRACTION (UNIT: METER)

w_b	w_v	h_g	d_c	w_s	σ_H	σ_Z	d_H	v_T	v_S
3	0.05	0.4	0.15	0.1	2	10	5	0.5	0.2

representing individual objects via Euclidean distance clustering and voxel-based normalized cut segmentation. Finally, using the 3-D object matching framework, 3-D light poles, traffic signposts, and bus stations were extracted from the segmented clusters. The 3-D object extraction results along with the ground truth for these four data sets are listed in Table III. Fig. 17 shows a visual example of a part of a point cloud from the SRS data set and its processing results in each processing step. The extracted 3-D light poles, traffic signposts, and bus stations are shown in Fig. 17(e). As shown by the box labeled #A in Fig. 17, some overlapped palm trees were grouped into the same cluster after Euclidean distance clustering. Such a cluster was well segmented into individual objects through voxel-based normalized cut segmentation. However, as shown by the box labeled #B in Fig. 17, use of the voxel-based normalized cut segmentation method failed to segment two small overlapping banyan trees located very close to each other. As shown in Fig. 17(e), by using the proposed 3-D object matching framework, light poles with and without attachments (e.g., traffic signs and advertising boards), different shapes of traffic signposts, and bus stations were correctly extracted. However, the clustering and segmentation results of the off-ground points greatly affect our proposed 3-D object extraction algorithm. Regarding light poles and traffic signposts that are hidden in the trees, our proposed clustering and segmentation methods fail to separate them from the trees. Therefore, the use of our proposed 3-D object extraction algorithm fails to extract such light poles and traffic signposts.

To quantitatively assess the accuracy and correctness of the 3-D object extraction results on these four data sets, the following four measures are used: *completeness*, *correctness*, *quality* [50], and F_1-*measure* [27]. *Completeness* depicts the

TABLE III
GROUND TRUTH AND EXTRACTION RESULTS

Dataset	Ground Truth			Extraction Result			
	Light Pole	Traffic Signpost	Bus Station	Light Pole	Traffic Signpost	Bus Station	False Alarm
SRS	267	81	11	253	78	11	9
YRS	154	53	4	144	49	4	10
ICEC	129	64	0	121	61	0	3
XR	135	55	6	129	54	6	5

percentage of true positives in the ground truth; *correctness* describes the percentage of true positives in the extraction result; *quality* and F_1-*measure* are two overall measures. These four measures are defined as follows:

$$completeness = \frac{TP}{TP + FN} \tag{20}$$

$$correctness = \frac{TP}{TP + FP} \tag{21}$$

$$quality = \frac{TP}{TP + FP + FN} \tag{22}$$

$$F_1 - measure = \frac{2 \cdot completeness \cdot correctness}{completeness + correctness} \tag{23}$$

where TP, FN, and FP denote the number of *true positives*, *false negatives*, and *false positives*, respectively. The quantitative evaluation results using these four measures are detailed in Table IV. The proposed 3-D object extraction algorithm achieves an average completeness, correctness, quality, and F_1-measure of 0.949, 0.971, 0.922, and 0.960, respectively. Therefore, our proposed 3-D object extraction algorithm is suitable for extracting 3-D light poles, traffic signposts, and bus stations from MLS point clouds.

Our proposed algorithm was implemented using C++ and executed on an HP Z820 8-core-16-thread workstation. The computing time in each processing step was recorded for these four data sets (see Table V). In practice, each of the data sets was first partitioned into a group of data segments with a road

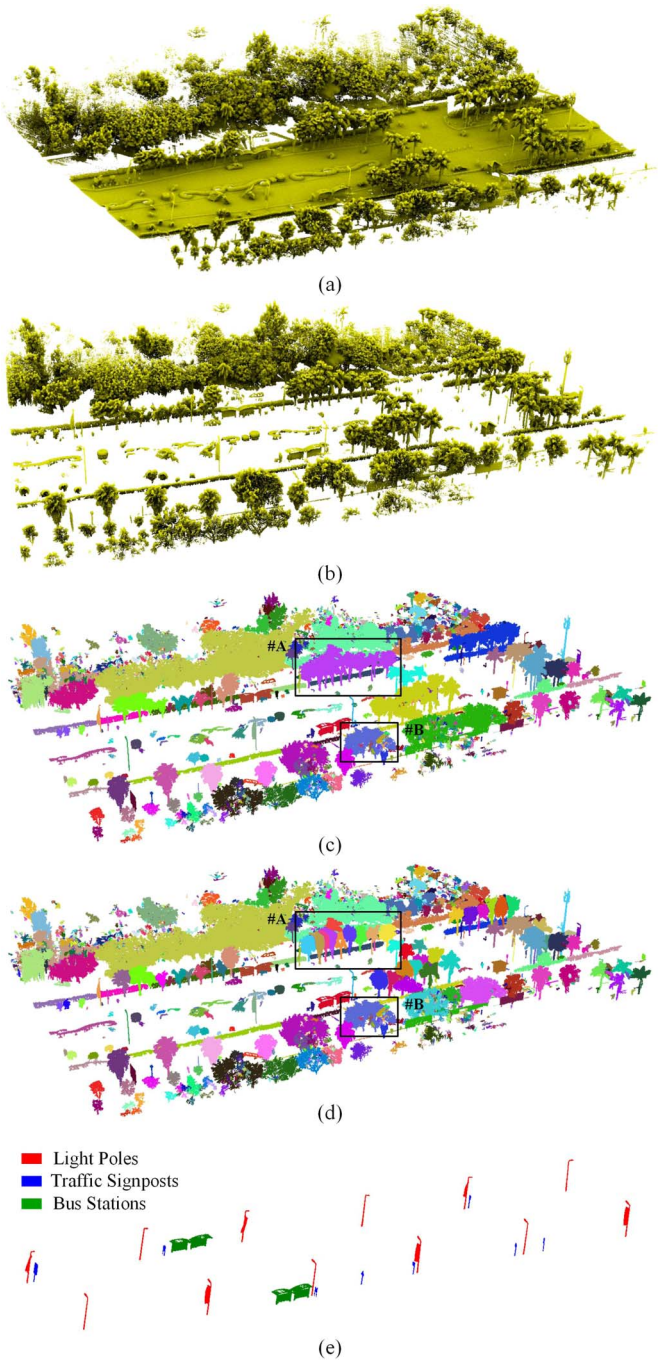


Fig. 17. (a) A part of a point cloud from SRS data set, (b) off-ground points, (c) clustered off-ground points, (d) segmented off-ground points, and (e) the extracted 3-D light poles, traffic signposts, and bus stations.

length of about 50 m. Then, all the segments were distributed to a multithread computing environment with 16 parallel threads. Thus, 16 segments were under process simultaneously. Such an organization dramatically improves the performance and reduces the time complexity of our proposed algorithm. As shown in Table V, the largest data set (SRS data set) containing about 581 million points and a road segment of approximately 4260 m was processed in 2115 s. Therefore, our proposed algorithm is suitable for rapidly handling large volumes of MLS point clouds for automated extraction of urban road facilities.

TABLE IV
QUANTITATIVE EVALUATION RESULTS

Dataset	Completeness	Correctness	Quality	F_1-Measure
SRS	0.953	0.974	0.929	0.963
YRS	0.934	0.952	0.891	0.943
ICEC	0.943	0.984	0.929	0.963
XR	0.964	0.974	0.940	0.969
Average	0.949	0.971	0.922	0.960

TABLE V
COMPUTING TIME ON THE FOUR DATA SETS (UNIT: SECOND)

Dataset	Ground Removal	Clustering & Segmentation	Object Extraction	Total
SRS	12	47	2056	2115
YRS	7	31	1068	1106
ICEC	8	33	1114	1155
XR	10	40	1816	1866

TABLE VI
GROUND TRUTH AND EXTRACTION RESULTS

Ground Truth			Extraction Result			
			Proposed		Percentile	
Light Pole	Clean	Attached	Light Pole	False Alarm	Light Pole	False Alarm
65	36	29	62	4	51	5

F. Comparative Studies

Comparative studies were also conducted to further compare our proposed algorithm with the percentile-based algorithm [34] in extracting 3-D light poles from MLS point clouds. A point cloud data set containing about 160 million points and with a length of about 948 m in the road direction was selected for the comparative studies. The ground truth and light pole extraction results obtained using different methods are detailed in Table VI. In Table VI, "Clean" denotes the number of light poles without any attachments; "Attached" denotes the number of light poles with attachments such as advertising boards and traffic signs. Fig. 18 shows a part of the point cloud from the selected data set along with the extracted 3-D light poles using the proposed algorithm and the percentile-based algorithm, respectively. To extract light poles, the percentile-based algorithm first horizontally divided an off-ground object into four quartiles. Then, to reduce the impact of the shrubs attached at the bottom of a light pole, as well as other attachments to the pole such as advertising boards and traffic signs, the third quartile was selected for recognizing pole-like structures. Finally, an off-ground object with a vertical pole-like structure in its third quartile was regarded as a light pole. However, as shown by the box in Fig. 18(b), in the selected data set, large-size advertising boards are attached to some of the light poles on their third quartiles. Thus, the percentile-based algorithm failed to recognize pole-like structures from the third quartile. As a result, light poles with such attachments failed to be extracted [see Fig. 18(c)]. Comparatively, by using the proposed 3-D object matching framework, our proposed algorithm extracted both

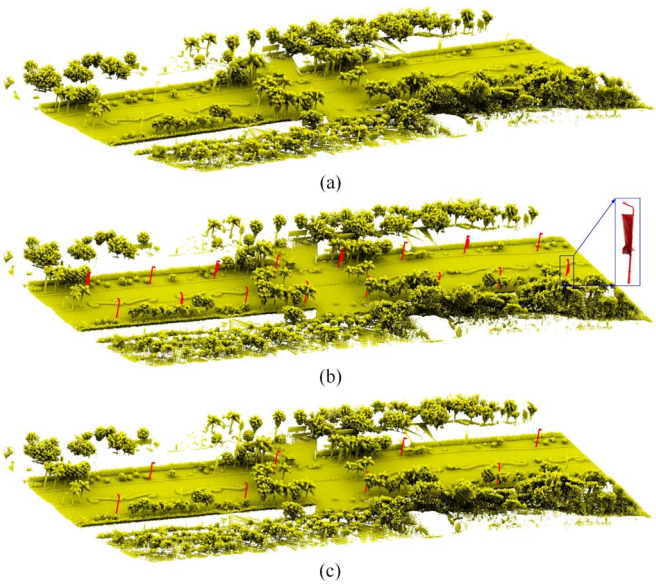


Fig. 18. (a) Raw point cloud, (b) and (c) extracted 3-D light poles using the proposed algorithm and the percentile-based algorithm, respectively.

TABLE VII
QUANTITATIVE EVALUATION RESULTS

Method	Completeness	Correctness	Quality	F_1-Measure
Proposed	0.954	0.939	0.899	0.946
Percentile	0.785	0.911	0.729	0.843

3-D light poles with and without attachments [see Fig. 18(b)]. In addition, quantitative evaluations were also performed on the light pole extraction results using completeness, correctness, quality, and F_1-meausre. The evaluation results are listed in Table VII. In conclusion, our proposed 3-D object extraction algorithm outperforms the percentile-based algorithm in correctly and completely extracting 3-D light poles from MLS point clouds.

V. CONCLUSION

In this paper, we have presented a novel 3-D object extraction algorithm for rapid, automated extraction of urban road facilities for transportation-related applications. Our proposed algorithm was successfully applied to four point cloud data sets for extracting 3-D light poles, traffic signposts, and bus stations directly from 3-D MLS data. Quantitative evaluations demonstrated that our proposed algorithm achieves an average completeness, correctness, quality, and F_1-measure of 0.949, 0.971, 0.922, and 0.960, respectively. By adopting a multithread computing strategy, our proposed algorithm rapidly handles large volumes of MLS point clouds toward 3-D object extraction. For a point cloud data set containing about 581 million points and a road direction distance of approximately 4260 m, the total computing time for extracting 3-D light poles, traffic signposts, and bus stations is within 36 min. Comparative studies also demonstrated that our proposed algorithm outperforms the percentile-based algorithm in extracting 3-D light poles. Therefore, we provide a feasible, promising solution to rapid, automated extraction of urban road facilities toward transportation-related applications using an MLS system. However, because the voxel-based normalized cut segmentation method cannot segment very closely overlapped object clusters, our proposed algorithm might encounter problems when extracting light poles and traffic signposts that are hidden in trees. In our future work, we will explore and develop new methods, which do not depend on point cloud segmentation operations, for object extraction and recognition from 3-D MLS point clouds.

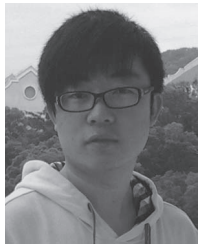
ACKNOWLEDGMENT

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

REFERENCES

[1] N. 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 technologies," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1403–1415, Sep. 2013.
[5] 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.
[6] 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.
[7] 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, 2013.
[8] C. K. Toth, "R&D of mobile LiDAR mapping and future trends," in *Proc. ASPRS Annu. Conf.*, Baltimore, MD, USA, 2009, pp. 1–7.
[9] H. S. Koppula, A. Anand, T. Joachims, and A. Saxena, "Semantic labelling of 3-D point clouds for indoor scenes," in *Proc. Adv. Neural Inf. Process. Syst.*, Granada, Spain, 2011, pp. 244–252.
[10] D. I. Kim and G. S. Sukhatme, "Semantic labelling of 3-D point clouds with object affordance for robot manipulation," in *Proc. IEEE Int. Conf. Robot. Autom.*, Pokfulam, Hong Kong, 2014, pp. 5578–5584.
[11] S. H. Khan, M. Bennamoun, F. Sohel, and R. Togneri, "Geometry driven semantic labeling of indoor scenes," in *Proc. Eur. Conf. Comput. Vis.*, Zürich, Switzerland, 2014, pp. 679–694.
[12] R. B. Rusu, A. Holzbach, N. Blodow, and M. Beetz, "Fast geometric point labelling using conditional random fields," in *Proc. Int. Conf. Intell. Robot. Syst.*, St. Louis, MO, USA, 2009, pp. 7–12.
[13] 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.
[14] 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.
[15] Z. Wang *et al.*, "A multiscale and hierarchical feature extraction method for terrestrial laser scanning point cloud classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2409–2425, May 2015.
[16] O. van Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or, "A survey on shape correspondence", *Comput. Graph. Forum*, vol. 30, no. 6, pp. 1681–1707, 2011.
[17] J. W. H. Tangelder and R. C. Veltkamp, "A survey of content based 3-D shape retrieval methods," *Multimedia Tools Appl.*, vol. 39, no. 3, pp. 441–471, Sep. 2008.
[18] J. Corney *et al.*, "Coarse filters for shape matching," *IEEE Comput. Graph. Appl.*, vol. 22, no. 3, pp. 65–74, May/Jun. 2002.

[19] E. Zucherberger, A. Tal, and S. Shlafman, "Polyhedral surface decomposition with applications," *Comput. Graph.*, vol. 26, no. 5, pp. 733–743, Oct. 2002.
[20] T. Funkhouser *et al.*, "A search engine for 3-D models," *ACM Trans. Graph.*, vol. 22, no. 1, pp. 83–105, 2003.
[21] O. van Kaick, H. Zhang, and G. Hamarneh, "Bilateral maps for partial matching," *Comput. Graph. Forum*, vol. 32, no. 6, pp. 189–200, Sep. 2013.
[22] Y. Zheng, C. L. Tai, E. Zhang, and P. Xu, "Pairwise harmonics for shape analysis," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 7, pp. 1172–1184, Jul. 2013.
[23] H. Tabia, M. Daoudi, J. P. Vandeborre, and O. Colot, "A new 3-D-matching method of nonrigid and partially similar models using curve analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 4, pp. 852–858, Apr. 2011.
[24] G. Pan *et al.*, "Establishing point correspondence of 3-D faces via sparse facial deformable model," *IEEE Trans. Image Process.*, vol. 22, no. 11, pp. 4170–4181, Nov. 2013.
[25] Q. Zhang, X. Song, X. Shao, R. Shibasaki, and H. Zhao, "Unsupervised skeleton extraction and motion capture from 3-D deformable matching," *Neurocomputing*, vol. 100, pp. 170–182, Jan. 2013.
[26] R. Litman and A. M. Bronstein, "Learning spectral descriptors for deformable shape correspondence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 1, pp. 171–180, Jan. 2014.
[27] H. Guan *et al.*, "Using mobile laser scanning data for automated extraction of road markings," *ISPRS J. Photogramm. Remote Sens.*, vol. 87, pp. 93–107, Jan. 2014.
[28] Y. Zhou, Y. Yu, G. Lu, and S. Du, "Super-segments based classification of 3-D urban street scenes," *Int. J. Adv. Robot. Syst.*, vol. 9, no. 248, pp. 1–8, Dec. 2012.
[29] R. Zhong, J. Wei, W. Su, and Y. F. Chen, "A method for extracting trees from vehicle-borne laser scanning data," *Math. Comput. Model.*, vol. 58, no. 3, pp. 733–742, Aug. 2013.
[30] G. Vosselman, "Slope based filtering of laser altimetry data," *Int. Archives Photogramm. Remote Sens.*, vol. 33, no. B3/2, pp. 935–942, 2000.
[31] K. Zhang *et al.*, "A progressive morphological filter for removing nonground measurements from airborne LiDAR data," *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 4, pp. 872–883, Apr. 2003.
[32] 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.
[33] 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.
[34] 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.
[35] 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.
[36] 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.
[37] 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, 2010.
[38] 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.
[39] D. Manandhai and R. Shibasaki, "Feature extraction from range data," in *Proc. Asian Conf. Remote Sens.*, Singapore, 2001, vol. 5, pp. 1–6.
[40] 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.
[41] 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.
[42] 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.
[43] A. Vu, Q. Yang, J. A. Farrell, and M. Barth, "Traffic sign detection, state estimation, and identification using onboard sensors," in *Proc. Int. IEEE Annu. Conf. Intell. Transp. Syst.*, The Hague, The Netherlands, 2013, pp. 875–880.
[44] 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.
[45] 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.
[46] E. Wahl, U. Hillenbrand, and G. Hirzinger, "Surflet-pair-relation histograms: A statistical 3-D-shape representation for rapid classification," in *Proc. Int. Conf. 3-D Digit. Imag. Model.*, Banff, U.K., 2003, pp. 474–481.
[47] H. Li, X. Huang, and L. He, "Object matching using a locally affine invariant and linear programming techniques," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 2, pp. 411–424, Feb. 2013.
[48] H. Jiang, M. S. Drew, and Z. N. Li, "Matching by linear programming and successive convexification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 959–975, Jun. 2007.
[49] M. Körtgen, G. J. Park, M. Novotni, and R. Klein, "3-D shape matching with 3-D shape contexts," in *Proc. Central Eur. Seminar Comput. Graph.*, Wien, Austria, 2003, vol. 3, pp. 5–17.
[50] M. Rutzinger, F. Rottensteiner, and N. Pfeifer, "A comparison of evaluation techniques for building extraction from airborne laser scanning," *IEEE J. Sel. Topics Earth Observ. Remote Sens.*, vol. 2, no. 1, pp. 11–20, Mar. 2009.



Yongtao Yu received the B.S. degree in computer science and technology from Xiamen University, Xiamen, China, in 2010. He is currently working toward the Ph.D. degree in computer science and technology in the Department of Computer Science, Xiamen University.

His current research interests include computer vision, machine learning, mobile laser scanning, and information extraction from 3-D point clouds.

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

He is currently a Professor with the School of Information Science and Engineering, Xiamen University, Xiamen, China. His current research interests include information extraction from earth observation images and 3-D surface reconstruction from mobile laser scanning point clouds.

Haiyan Guan received the Ph.D. degree in geomatics from University of Waterloo, Waterloo, ON, Canada, in 2014.

She is currently an Associate Professor with the College of Geography and Remote Sensing, Nanjing University of Information Science and Technology, Nanjing, China. Her current 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) received the Ph.D. degree in information communication engineering from the National University of Defense Technology, Changsha, China, in 2002.

He is currently a Professor with the School of Information Science and Engineering, Xiamen University, Xiamen, China. His current research interests include remote sensing image processing, mobile laser scanning data analysis, and multisensor fusion.