3-D Road Boundary Extraction From Mobile Laser Scanning Data via Supervoxels and Graph Cuts

Dawei Zai, Jonathan Li, *Senior Member, IEEE*, Yulan Guo, *Member, IEEE*, Ming Cheng, *Member, IEEE*, Yangbin Lin, Huan Luo, and Cheng Wang, *Senior Member, IEEE*

***Abstract*— Effective extraction of road boundaries plays a significant role in intelligent transportation applications, including autonomous driving, vehicle navigation, and mapping. This paper presents a new method to automatically extract 3-D road boundaries from mobile laser scanning (MLS) data. The proposed method includes two main stages: supervoxel generation and 3-D road boundary extraction. Supervoxels are generated by selecting smooth points as seeds and assigning points into facets centered on these seeds using several attributes (e.g., geometric, intensity, and spatial distance). 3-D road boundaries are then extracted using the α-shape algorithm and the graph cuts-based energy minimization algorithm. The proposed method was tested on two data sets acquired by a RIEGL VMX-450 MLS system. Experimental results show that road boundaries can be robustly extracted with an average completeness over 95%, an average correctness over 98%, and an average quality over 94% on two data sets. The effectiveness and superiority of the proposed method over the state-of-the-art methods is demonstrated.**

***Index Terms*— Mobile Laser Scanning (MLS), supervoxel, road boundary, road detection, point cloud, graph cuts.**

I. Introduction

As A transportation infrastructure, roads play a significant role in a variety of applications including driver assistance and safety warning systems [1], [2], autonomous driving [3] and vehicle navigation [4]. Accurate and up-to-date road information is a critical requirement for intelligent transportation system updating and provides auxiliary information for intelligent vehicle applications to make decisions and improve driving safety. Besides, roads provide rich contextual cues for the detection and maintenance of road facilities (e.g., light poles, traffic signposts, road markings, manholes, and sewer well covers). Consequently, it is important and also challenging to effectively extract roads from surveyed data, e.g., satellite and aerial imagery, Airborne Laser Scanning (ALS) data, and Mobile Laser Scanning (MLS) data. This paper proposes a method to extract 3D road boundaries from MLS data for creating high-definition road maps that can be used for navigating driverless cars.

MLS systems have recently attracted increasing attention in the areas of transportation, navigation and autonomous driving. A MLS system usually consists of laser scanners, an integrated Global Navigation Satellite System/Inertial Measurement Unit (GNSS/IMU) system, and several digital cameras. It produces three-dimensional (3D) point clouds of a scene by recording its geometry and intensity information, and records color/texture information produced by digital cameras. Kaartinen *et al.* [5] tested the performance of various MLS systems (i.e., ROAMER, RIEGL VMX-250, Sensei, Streetmapper 360, and Optech Lynx) by collecting laser point clouds from a given field and then verifying the point cloud precision under favorable GNSS conditions. Guan *et al.* [6] tested the performance of an MLS system (RIEGL VMX-450) and developed algorithms to extract three road features including road surfaces, road markings, and pavement cracks. It is demonstrated that MLS is a reliable and cost-effective alternative for road inspection [6]. In addition, a number of methods have been proposed for MLS point cloud interpretation, such as pole extraction [7], traffic sign extraction [8], road marking extraction [9], point classification [10], and point labeling [11]. Therefore, MLS systems have great potential for road inspection, digital road mapping and city modelling.

In this paper, we propose a new method to extract 3D road boundaries from MLS data in complex urban environments. The proposed paper includes two main stages: supervoxel generation for segmentation, and the alpha-shape algorithm and graph cuts for road boundary extraction. The remainder of this paper is organized as follows. Section 2 reviews and discusses related methods for road extraction from surveyed data. Section 3 introduces the proposed road boundary extraction method, including supervoxel generation and 3D road boundary extraction. Section 4 presents the experimental results and discussion. Section 5 concludes the paper.

II. Related Work

To extract roads from satellite and aerial images, Quackenbush [12] and Mena [13] reviewed several

Manuscript received October 9, 2016; revised February 23, 2017; accepted April 27, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant 41471379 and in part by the Fujian Collaborative Innovation Center for Big Data Applications in Governments. The Associate Editor for this paper was S. S. Nedevschi. *(Corresponding author: Jonathan Li.)*

D. Zai, M. Cheng, H. Luo, and C. Wang are with Fujian Key Laboratory of Sensing and Computing for Smart City, School of Information Science and Engineering, Xiamen University, Xiamen 361005, China.

J. Li is with the Fujian Key Laboratory of Sensing and Computing for Smart City, School of Information Science and Engineering, Xiamen University, Xiamen 361005, China, and also with WatMos Lab, Department of Geography and Environmental Management, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: junli@xmu.edu.cn).

Y. Guo is with the College of Electronic Science and Engineering, National University of Defense Technology, Changsha HN 410073, China.

Y. Lin is with the Computer Engineering College, Jimei University, Xiamen 361005, China.

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.2017.2701403

representative methods. These methods mainly focus on rural areas, where road networks are relatively sparse and regular. Due to challenges including occlusions, shadows and cars, it is difficult to extract roads from passive remotely sensed imagery in complicated dense scenes, with only limited methods being proposed in this direction [14].

A number of methods have recently been proposed to perform road extraction from ALS data. Alharthy and Bethel [15] obtained road points using intensity-based filtering and then extracted road networks using connected components. Choi *et al.* [16] used both height and reflectance information to extract road points. Vosselman and Zhou [17] detected curbstones from ALS data using small height jumps near the terrain surface and generated the midpoints of high and low points on both sides of the height jump. They then used these midpoints to fit a smooth curve. Zhou and Vosselman [18] extended this work to extract curbstones in both ALS and MLS data. Boyko and Funkhouser [19] produced a map spline by projecting a 2D map onto the 3D point cloud merged from multiple airborne and mobile laser scans of an urban environment. They then divided the road network into patches using the map spline. By optimizing an active contour, they developed a classical ribbon snake algorithm for each patch. Finally, they labeled the points falling inside the active contour as road points. Hu *et al.* [20] initially classified the point clouds into ground and non-ground points. They then extracted the road center lines using mean shift, tensor voting and Hough transform.

Compared to ALS systems, MLS systems can acquire high density and accurate point clouds along roads over large areas. MLS data is more suitable for road extraction in several aspects: (1) Since the platform for the MLS system moves on a road, occlusions of the road caused by buildings and trees are reduced significantly. (2) As an active sensing technique, there is no shadow in MLS data. (3) The trajectory data produced by a MLS system can provide accurate and real-time position information of a vehicle, it also provides location and direction information of the underlying road of the vehicle. (4) Since road surfaces have similar reflectance, the intensity information of MLS data can be used as an additional feature for road extraction. Consequently, road extraction from MLS data has become an active research topic in recent years. These methods can be roughly classified into two categories: 1) projection-based methods that extract roads from range images; 2) 3D-based methods that extract roads using features from point clouds.

A. Projection-Based Methods

Generally, the projection-based methods initially generate range images using various attributes of MLS data (e.g., height, intensity and pulse width) [21]–[23]. Jaakkola *et al.* [21] applied image processing algorithms (e.g., cropping, fitting and filtering) to the raster images created from MLS data based on intensity and height features to detect curbs. Hernández and Marcotegui [22] firstly filtered artifacts in range images, which were generated by projecting 3D points onto a plane. They then used a quasi-flat zone algorithm and a region adjacency graph representation to extract the contour between pavements and roads. Kumar *et al.* [23] used the height, reflectance and pulse width attributes to generate 2D raster images. From these images, road boundaries were extracted using a combination of two modified parametric snake model methods. Serna and Marcotegui [24] mapped point clouds to range images and then detected curb candidates using both height and elongation. However, with these methods, unnecessary matching errors are introduced in the rasterizing process and it is difficult to obtain accurate results of road boundaries.

B. 3D-Based Methods

The 3D-based methods use road features, such as smooth surfaces/polynomials and local patterns (e.g., curbs), to detect road boundaries. Several 3D-based methods use smooth surface or polynomial to detect road boundaries. Smadja *et al.* [25] used a RANSAC algorithm to generate a polynomial representation for the road in MLS data. Yuan *et al.* [26] employed a maximum entropy based fuzzy clustering method to cluster points and a weighted linear fitting algorithm to generate road surfaces. These smooth surface/polynomial based methods are sensitive to the percentage of road surface points in each data block and may lose some road details (e.g., corners and undulant areas). Other 3D-based methods use local patterns to detect road boundaries. Ibrahim and Lichti [27] employed a derivative of the Gaussian filtering algorithm to detect curbs from MLS data. Yang *et al.* [28] first used GPS time of points to divide point clouds into sequential road cross sections. They then used a moving window operator to detect curb points in each section based on elevation differences, point density and slope changes. Guan *et al.* [6] used trajectory data to partition point clouds into a number of blocks. Both slope and elevation tests were used to detect curb point in each block. Wang *et al.* [29] constructed a saliency feature map on point clouds to detect curb points. Zai *et al.* [30] used local linear characteristic of road boundary to detect curb points with graph cuts. Because these local pattern based methods were developed to extract road boundaries using curbs in urban environments, they may face great challenges in real applications such as irregular curbs and grass strips surrounding the curbs.

In this paper, we propose a new method to extract 3D road boundaries from MLS data. Our method consists of two main stages: supervoxel generation and 3D road boundary extraction. An illustration of the workflow of our method is shown in Fig. 1. In the first stage, the method first selects smooth points as seeds and then assigns point clouds into facets centered on these seeds based on the attributes of the points (e.g., geometric, intensity and spatial distance). The second stage contains three steps: (1) The boundary points are extracted using the α-shape algorithm. (2) Graph cuts based energy minimization are used to extract the road boundary points. (3) Refinement is used to remove outliers based on Euclidean distance clustering, to merge the road boundaries and fit them into smooth curves. We tested our proposed method on two datasets acquired by a RIEGL

VMX-450 MLS system. The results have demonstrated the effectiveness and feasibility of our proposed 3D road boundary extraction method. Zai *et al.* [30] evenly partitioned point clouds into voxels to remove non-ground points and terrain points and then extracted road boundaries using graph cuts based on the local linear characteristics of road boundary. This paper is an extension of [30], its contributions are as follows: 1) We propose a new method to automatically extract 3D road boundaries from large-scale 3D point clouds; 2) We propose an improved supervoxel algorithm to segment road surfaces into facets; and 3) We propose a graph cuts based energy minimization algorithm to effectively extract road boundaries using trajectory data as an original observation model.

III. Road Boundary Extraction

Due to the large number of points, it is highly challenging to process a large-scale point cloud. Therefore, it is necessary to partition point clouds into several blocks before further processing. In this paper, the trajectory of the vehicle, on which the MLS system is mounted, is used to partition the point clouds into a number of overlapping blocks. These blocks are segmented using a predefined distance interval along the trajectory.

For each block, supervoxels are first generated according to the points' attributes (e.g., geometric, intensity) and spatial distances between the points. A set of facets are then obtained by integrating neighboring coplanar points. Next, the boundary points between each pair of adjacent non-coplanar facets are extracted. Road boundary points are then extracted using the graph cuts based energy minimization algorithm. Finally, outliers are removed based on Euclidean distance clustering and the road boundary points are merged and fitted into smooth curves.

A. Supervoxel Generation

Our proposed method focuses on road boundary extraction from MLS point clouds. To extract road boundaries, an intuitive strategy is to extract road surfaces first. To extract planes as large as possible, existing plane extraction algorithms tend to under-segment point clouds. Examples of point cloud under-segmentation methods include expectation maximum [31], Hough transform [32], RANSAC [33], and region growing [34]. However, these algorithms cannot accurately extract road boundaries for rough roads or roads occluded by cars or pedestrians. To obtain road boundaries as complete as possible, over-segmentation strategy is used to divide the point clouds into facets (supervoxels). Supervoxel over-segments point clouds into regions, within which points have similar properties, while maintaining boundary information as much as possible. Inspired by the 3D point cloud segmentation work [35], we propose an improved supervoxel algorithm, to segment road surfaces into a set of facets. Lin *et al.* [35] outperforms the state-of-the-art algorithms in two major aspects, which are beneficial for our road boundary extraction.

1. Several existing methods select supervoxel cluster seeds by evenly partitioning 3D space with a fixed size R_{seed} [10], [36]. Ramiya *et al.* [37] obtained supervoxels using Voxel Cloud Connectivity Segmentation (VCCS) [36]. Babahajiani *et al.* [38] generated supervoxels by evenly partitioning point clouds into voxels and merging voxels via region growing. Song *et al.* [39] generated supervoxels through two steps: detecting boundaries frame by frame and clustering points by first excluding the boundary points. However, it is time-consuming for high-dense point clouds. Our algorithm selects saliently smooth points as seeds to ensure that structures (e.g. curbs) smaller than R_{seed} can be well preserved.

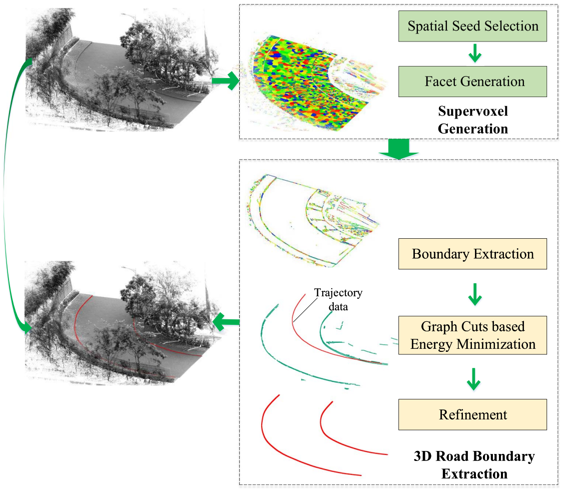


Fig. 1. The workflow of our 3D road boundary extraction method.

2. Several existing methods generate voxels using a fixed size [36] or two different sizes [10]. Our algorithm generates voxels with adaptive sizes. Specifically, for MLS point clouds with uneven density, some boundaries can be lost with large voxels since too many neighboring points are segmented into one voxel. In contrast, voxels with small sizes may be too sparse to connect with their neighbors. Therefore, it is difficult to achieve good segmentation performance using a fixed-size voxel generation method. However, adaptive size voxel generation method can avoid these problems.

While whole point clouds are segmented into facets [35], our method segments road surfaces into facets and removes uninterested points to extract road boundaries.

1) Spatial Seed Selection: Our algorithm starts by computing the tangent plane and smoothness for each point p_i of a given point cloud P. The tangent plane $T(p_i)$ of p_i is represented as two-tuples:

$$T(p_i) = (o_i, \vec{n_i}) \tag{1}$$

where o and $\vec{n}$ are the center and the normal vector of p_i, respectively. They are calculated from the $K-$neighborhood $N_K(p_i)$ of p_i.

The iterative weighted least square algorithm is used to generate the tangent plane for p_i. For each point p, within the K-neighborhood of p_i, the weight $w(p)$ is computed as:

$$w(p) = \begin{cases} (1 - (\frac{\text{dist}(p,\ T^n(p_i))}{\epsilon})^2)^2, \\ \quad \text{dist}(p, T^n(p_i)) < \epsilon \\ 0, \quad \text{dist}(p, T^n(p_i)) \geq \epsilon \end{cases} \tag{2}$$

$$\mathrm{dist}(p,\ T^n(p_i)) = |(p - o_i) \cdot \vec{n_i}| \tag{3}$$

where n denotes the iterative number ($n = 0, 1, 2, \ldots$) and ϵ is a predefined threshold to determine if point p belongs to the tangent plane $T^n(p_i)$. Then the tangent plane $T(p_i)$ is obtained by iteratively solving Eq. 4.

$$\underset{T(p_i)}{\operatorname{argmin}} \sum_{p \in N_K(p_i)} w(p) * \mathrm{dist}(p, T(p_i))^2 \tag{4}$$

This process is repeated for each point until either the algorithm is converged or a fixed number of iterations (3 in this paper) is reached. The smoothness $s(p_i)$ for a point p_i is then calculated as:

$$s(p_i) = \frac{\lambda_2}{\lambda_3} \tag{5}$$

where λ_1, λ_2 and λ_3 ($\lambda_1 \geq \lambda_2 \geq \lambda_3$) are the eigenvalues of the covariance matrix of the points used to compute the tangent plane $T(p_i)$.

Once the smoothness $s(p_i)$ of each point p_i is obtained, points with high smoothness are selected as initial candidates for facet initialization since these points have a higher possibility to be expanded to a facet.

Since our work focuses on road boundary extraction, the following two constraints are used to remove non-ground candidate points, which are different from [35]:

(a). Points higher than the road surface are removed from the candidate list;

(b). The angle between the normal of a candidate's tangent plane and the Z-axis must be smaller than a predefined threshold θ.

Finally, the candidate list Q_c is represented as:

$$\begin{aligned} Q_c = \{p \in P\ |s(p_i) > s(p_j) \text{ if } i < j \\ \wedge\ p.z - p_l.z < h\ \wedge\ T(p).\vec{n} \cdot \vec{z} \geq cos\theta\} \end{aligned} \tag{6}$$

where $s(p_i) > s(p_j)$ if $i < j$ means that Q_c is sorted according to the smoothness. $p.z - p_l.z < h$ means that a point is removed from Q_c if its height difference with the lowest point p_l in the block is larger than a threshold h, where h should be large enough to retain all road surface points. $T(p)\vec{n} \cdot \vec{z} \geq cos\theta$ ($\vec{z} = (0, 0, 1)$) means a point is removed if the angle between its normal and the Z-axis is larger than θ.

2) Facet Generation: A facet f_i is defined as a triple:

$$f_i = (P_i, o, \vec{n}) \tag{7}$$

where P_i denotes a point set belonging to f_i, o and $\vec{n}$ are the center and the normal of f_i, respectively. Given a candidate point cp_i from the candidate list Q_c, the initial facet is generated as $f_i = (\{cp_i\}, T(cp_i).o, T(cp_i).\vec{n})$. Within a neighborhood defined by a radius R_{seed} and the center cp_i, each point p is added to the point set $f_i.P_i$ and is removed from Qc if the following two requirements are satisfied: (a) The angle between $T(p).\vec{n}$ and $T(cp_i).\vec{n}$ is smaller than θ; and (b) The distance from p to $T(cp_i)$ is smaller than ϵ. If $f_i.P_i$ does not increase further, it is used to update $f_i.o$ and $f_i.\vec{n}$. Thus, a facet set is obtained when all candidates in Q_c have been tested.

Assigning points to facets is achieved using a distance measure calculated in a feature space (including spatial extent, normals and intensity). Since the proposed framework is generic, other features or descriptors as listed in [40] can also be added to the representation (e.g. FPFH). However, we found that the features used in this paper are sufficient to achieve high performance in our tasks.

Assigning points to facets ensures that the distance from a point to its facet is smaller than the distance to other facets. The distance function is calculated as:

$$D(p, f) = \sqrt{w_s * {D_s}^2 + w_n * {D_n}^2 + w_i * {D_i}^2} \tag{8}$$

where D_s, D_n, and D_i represent the Euclidean distance, normal distance and normalized intensity distance from a point to its facet, respectively. In this paper, we empirically set the spatial weight $w_s = 1$, the normal direction weight $w_n = 4$ and the intensity weight $w_i = 1$.

When an assignment is completed, the center of a facet's point set and the mean of the normal vectors of the tangent planes of all points are used to update the facet. This updating process is repeated until the facets do not change, or a fixed number of iterations is reached. To demonstrate the superiority of our proposed supervoxel algorithm for road boundary preservation, we conducted a comparative study (as shown in Fig. 2). Figures 2(b) and 2(c) show the results obtained by region growing [34]) and Voxel Cloud Connectivity Segmentation (VCCS) [36], respectively. The traditional region growing algorithm tends to under-segment point clouds to extract large planes, resulting in significant loss of boundary information. VCCS is unable to detect the boundary of some small structures because VCCS generates supervoxels with almost the same size. As shown in Fig. 2(d), our proposed algorithm segments road surfaces into a set of facets with adaptive sizes, successfully detects the boundaries and removes uninterested points.

B. 3D Road Boundary Extraction

1) Boundary Extraction: When point clouds are segmented into facets, the α-shape algorithm [41] is used to extract boundary points of facets. Next, the boundary points between each pair of adjacent co-planar facets are removed, as shown by red points in Fig. 3 (b) and the boundary point extraction results are obtained, as shown in Fig. 3 (c). It can be seen from Fig. 3 (d) that each boundary point has the same label as its facet.

2) Graph Cuts Based Energy Minimization: The extracted boundaries P_b include both road boundary points and other boundary points. Since the trajectory data provide the driving route of the vehicle on which the MLS system is mounted, the trajectory gives the approximate location of the road. For each boundary point bp, the minimum distance to the trajectory data Tra is computed as follows:

$$md = \min_{t_i \in Tra} (|bp - t_i|) \tag{9}$$

These boundary points are removed if $md > W$, where W is a given threshold and should be large enough to retain the road boundaries.

To generate road boundaries from the extracted boundaries, energy minimization is performed using an iterative graph-cuts

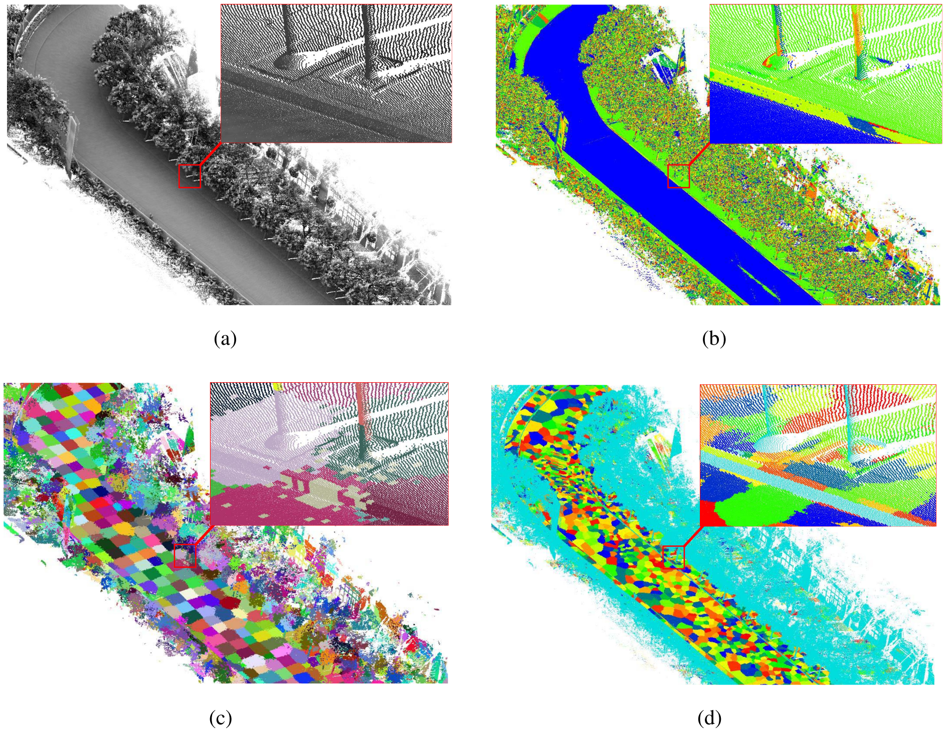


Fig. 2. An illustration of point cloud segmentation results achieved by different algorithms. (a) Original point cloud. (b) Results achieved by region growing. (c) Results achieved by VCCS. (d) Results achieved by the proposed algorithm (The removed points are shown in cyan).

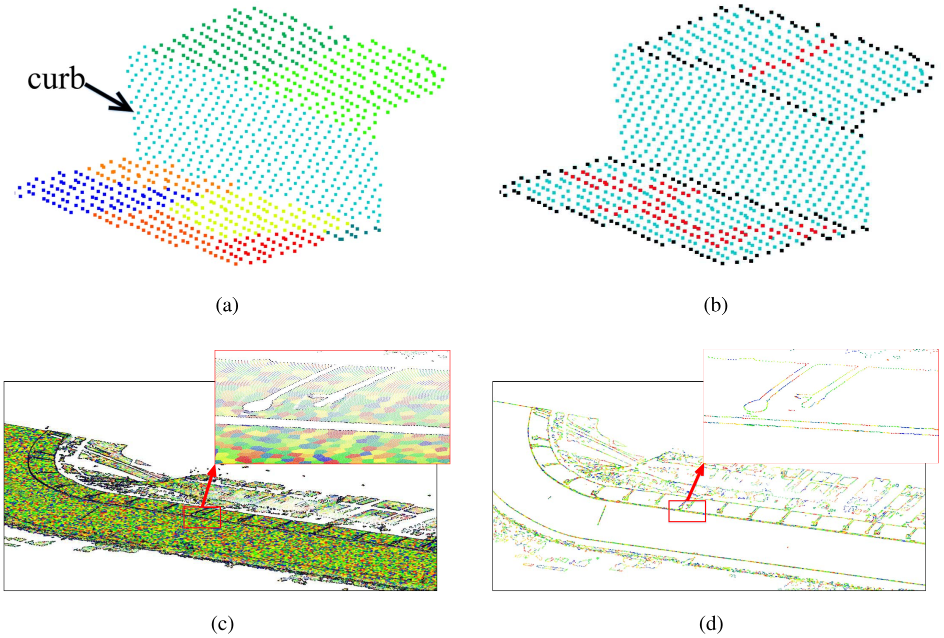
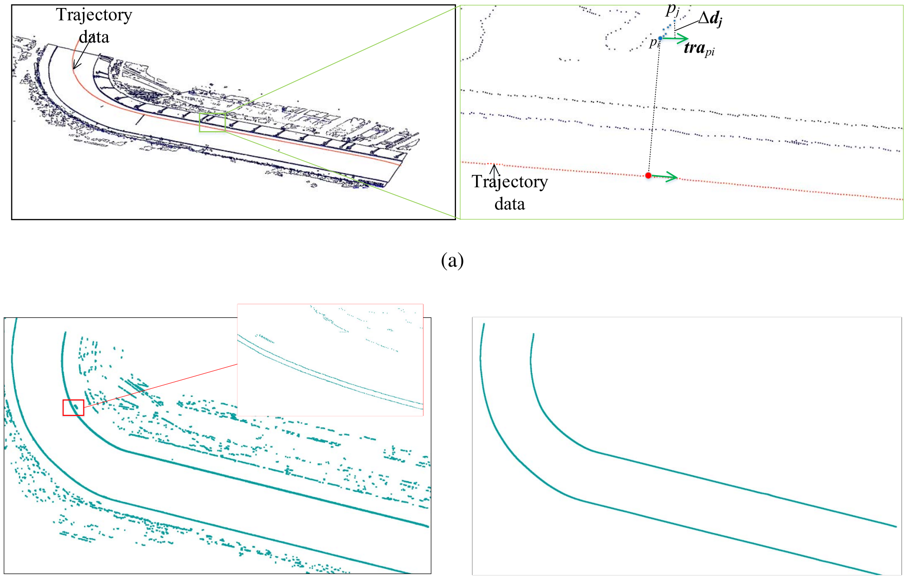


Fig. 3. Result of boundary points extraction. (a) Facets after segmenting a given section of point cloud (different facets are shown in different colors). (b) Extracted boundary points are shown in black and removed boundary points are shown in red. (c) Boundary points are shown in black. (d) The extracted boundary points (different colors represent different labels).

based algorithm [42]. The separation between road boundaries and non-road boundaries is actually a binary labelling problem.

Let $G =<V, E>$ be a weighted undirected graph, where each vertex $v_i \in V$ corresponds to a boundary point $p_i \in P_b$ and each undirected edge $e_{i,j} \in E$ represents a link between neighboring boundary point $p_i, p_j \in P_b$. The graph contains two additional vertices (terminals) V_s and V_t. Additional edges are created by connecting each boundary point $p_i \in P_b$ with the two terminal vertices, e_{i,V_s} and e_{i,V_t}. For a weighted graph, each edge $e \in E$ has an associated weight w_e. A cut $C \subset E$



(a)

(b) (c)

Fig. 4. Result of 3D road boundary extraction. (a) The trajectory data is shown in red. $\mathbf{tra(p_i)}$ is the direction of a trajectory point (the nearest one to p_i). Δd_j is the distance residual between p_j and the line defined by p_i and $\mathbf{tra(p_i)}$. (b) Result achieved by graph cuts based energy minimization. (c) Result achieved by curve fitting.

TABLE I
RIEGL VMX-450 PERFORMANCE

Laser scanner (VQ-450)	
Accuracy	$8\ mm$
Precision	$5\ mm$
Scan rate	$200\ lines/sec$
Measurement rate	$550,000\ pts/sec$
Minimum range	1.5 m
IMU/GPS (with DMI option)	
Position (absolute)	$2-5\ cm$
Position (relative)	$1\ cm$

See http://www.riegl.com/ for more details.

partitions the vertices V of the graph G into two disjoint sets S and T, where $V_s \in S$ and $V_t \in T$. The cost of each cut C is the sum of the weighted edge $e \in C$:

$$|C| = \sum_{e \in C} w_e \tag{10}$$

The minimum cut problem is defined as minimizing the cost for a binary labelling function f in order to produce the optimal segmentation with energy minimization. In this paper, we use the energy function proposed in [43].

This energy function plays an important role in binary labelling. Each point $p_i \in P_b$ is assigned a unique label in the label set L to represent road boundary or non-road boundary. The goal of this separation is to determine a labelling function f to assign each point $p_i \in P_b$ a label $f_{p_i} \in L$, where f is piecewisely smooth and has a low variation with respect to the observation data. It is observed that the driving route and the direction of the inspected road are provided by the trajectory data. That is, the direction of the road boundaries should be consistent with the direction of their nearest trajectory. Given a trajectory point tp_i and its four nearest neighboring trajectory points, the least square algorithm is used to calculate the direction vector of the point tp_i. In this paper, trajectory data is used as the preliminary observation model. The energy function is defined as:

$$E(f) = E_{data}(f) + \lambda \cdot E_{smooth}(f) \tag{11}$$

E_{smooth} measures the piecewise smoothness of f, E_{data} measures the variation between f and the observation data, and λ is a weighting parameter. E_{data} is computed as

$$E_{data}(f) = \sum_{p_i \in P_b} d(f_{p_i}) \tag{12}$$

where $d(f_{p_i})$ measures the extent to which the label f_{p_i} fits the point p_i in the observation data and is calculated as:

$$d(f_{p_i}) = \begin{cases} 1 - e^{(-\frac{\Delta d_i}{\sigma_1})} \\ \text{if } p_i \text{ is labeled as road boundary } (p_i \in S), \\ e^{(-\frac{\Delta d_i}{\sigma_1})} \\ \text{if } p_i \text{ is labeled as non-road boundary } (p_i \in T). \end{cases} \tag{13}$$

$$\Delta d_i = \frac{1}{n} \sum_{p_j} \Delta d_j \tag{14}$$

where n is the number of the points with the label of p_i, Δd_j is the residual distance from a point p_j with the same label as p_i to the line defined by p_i and the direction vector $\mathbf{tra(p_i)}$.

TABLE II
PARAMETER SETTING OF THE PROPOSED ALGORITHM USED ON TWO DATA SETS

Dataset	Description	Industrial Region	Ring Road
ϵ	A distance threshold to determine if a point lies on a facet	0.02	0.01
θ	An angle threshold to determine if two vectors are parallel	22.5°	22.5°
R_{seed}	The seed resolution used to constrain the radius of facets	0.3	0.15
K	The number of nearest neighbors	15	15
λ	The weighting parameter	128	32

As shown in Fig. 4 (a), $\mathbf{tra(p_i)}$ is the direction vector nearest to p_i in the trajectory points, σ_1 represents the average residual of all points.

As defined in Eq. 14, the weights of two edges e_{i,V_s} and e_{i,V_t}, are complementary. Among the two edges e_{i,V_s} and e_{i,V_t}, the one with a larger weight is cut during segmentation. More specifically, a road boundary point p_i has a small value of Δd, resulting in a small weight of e_{i,V_s} and a large weight of e_{i,V_t}. Furthermore, the definition of the data term (Eq. 12) favors the cut to go through e_{i,V_t} so that the point p_i is labelled as "road boundary". This definition of data term penalizes these road boundary points being labeled "non-road boundary", and vice versa. Therefore, the desired solution is enforced to be consistent with road boundary.

The smoothness term provides a measure for the difference between two neighboring boundary points $p_i, p_j \in P_b$ $(i \neq j)$ with labels f_{p_i}, f_{p_j}. Considering the pairwise relationship of data points, E_{smooth} is defined as

$$E_{smooth}(f) = \sum_{\{p_i,p_j\}\in N} B_{\{p_i,p_j\}} \cdot \delta(f_{p_i}, f_{p_j}) \tag{15}$$

where $\delta(\cdot)$ is set to 1 if $f_{p_i} \neq f_{p_j}$. Otherwise, $\delta(\cdot)$ is set to 0. $\delta(\cdot)$ is used to measure the discontinuity along segment boundaries. $B_{\{p_i,p_j\}}$ is a pairwise relationship function.

$$B_{\{p_i,p_j\}} = e^{-\frac{D_{p_i p_j}}{\sigma_2}} \tag{16}$$

where

$$D_{p_i p_j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \tag{17}$$

σ_2 denotes the spatial resolution of the point set P_b. As defined in Eqs. 16 and 17, close points with different labels have larger smoothness penalty values, while, the smoothness value is small for distant points, the smoothness penalty term is zero for points with the same label. That is, the smoothness term favors the neighboring points sharing the same label and penalizes the neighboring points with different labels. E_{data} and E_{smooth} are combined in the energy function to differentiate road boundary and non-road boundary points using the graph cuts based energy minimization method. Figure 4 (b) shows an illustration of the extracted road boundary points.

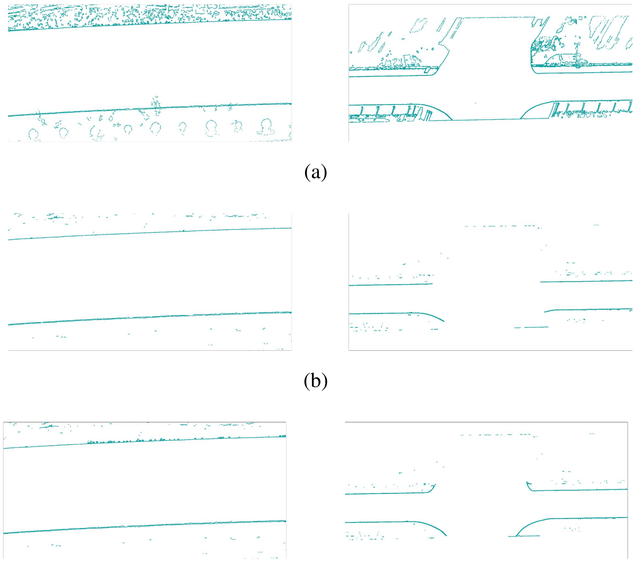

(a)
(b)
(c)

Fig. 5. Results achieved by the graph cuts based energy minimization method with different values of λ. (a) The extracted boundary points. (b) Extracted road boundaries with $\lambda = 32$. (c) Extracted road boundaries with $\lambda = 128$.

3) Refinement: It is observed that some outliers with similar directional characteristics as the road boundary might be incorrectly labeled as road boundary points. Meanwhile, curb boundaries are recognized as road boundaries [44] and have two parallel lines, as shown in Fig. 4 (b). However, we just consider the points on road surface as road boundaries. Therefore, a local clustering algorithm is used to extract road boundaries in each given point's neighborhood. First, points with lower height are selected as candidates for clustering since these points have a higher possibility to be considered as road boundaries. Given a candidate p_c, we remove it from the candidate list as outliers if the number of points within p_c's neighborhood is too small. Otherwise, all points within p_c's neighborhood are clustered into two clusters. The points within the cluster with an lower average height are considered as road boundaries.

Consequently, the detected road boundary points are sparsely distributed. A curve fitting algorithm [45] is finally used to generate smooth road boundaries for each road. Fig. 4 (c) shows an illustration of the refinement results.

IV. RESULTS AND DISCUSSION

A. MLS Data

The MLS data used in this paper were acquired by a RIEGL VMX-450 system, which includes two full-view RIEGL VQ-450 laser scanners, an inertial measurement unit (IMU), a global positioning system (GPS), and a distance measurement indicator (DMI). Table I shows the datasheet properties of all the sensors. Each laser scanner provides low-noise and gapless 360° lines at a measurement rate of 550, 000 pts/sec and a scan rate of up to 200 $lines/sec$. The typical absolute

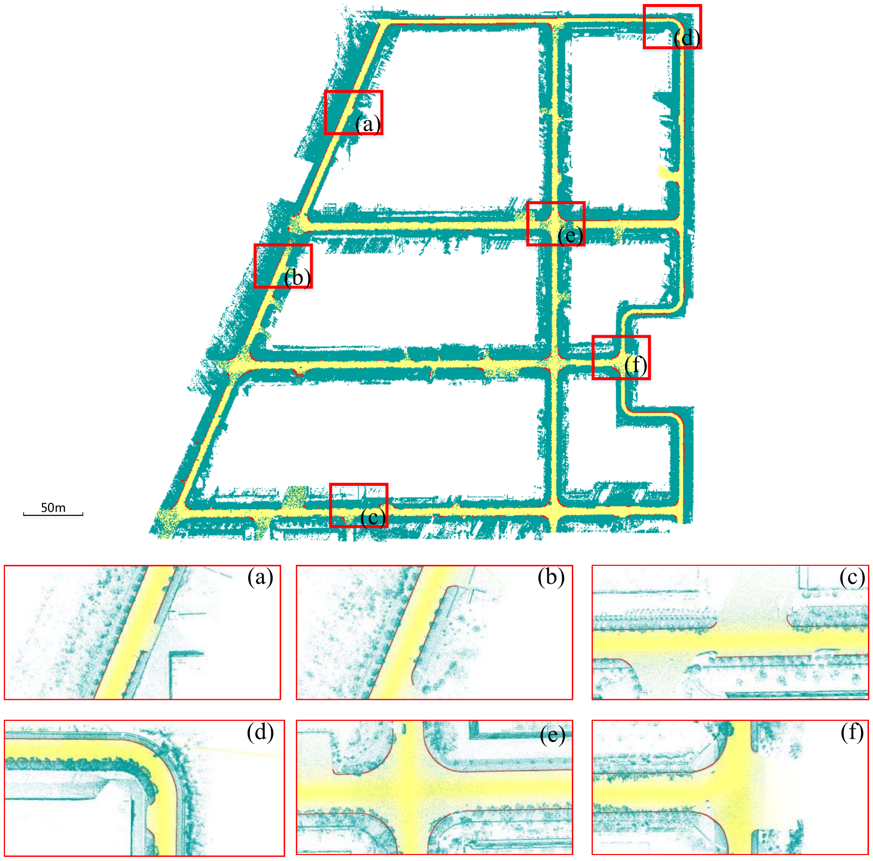


Fig. 6. Road boundary extraction results on the industrial region dataset. Crossroads and Intersections are shown in Figs. (d-f).

positioning accuracy of IMU/GPS with DMI option is typical $2-5\ cm$, and the relative positioning accuracy is $1\ cm$. The accuracy of the acquired point clouds is about $8\ mm$, and the precision is $5\ mm$. Two datasets covering an industrial region and a typical coastal ring road in Xiamen were selected to test 3D road boundary extraction performance. The industrial region dataset includes about 331 million points and covers an area of $828m \times 792m$. For the industrial region dataset, the length of the road boundaries is $9,292.79m$, and the length of the road is about $5000m$. The coastal ring road is a two directional road with four lanes. The coastal ring road dataset contains about 220 million points of a complex road scene and has a road distance of approximately $2000m$. Besides, this coastal ring road dataset includes irregular road boundaries of $3,939.03m$ with grass strips, trees, fences and cars, which pose significant challenges for road boundary extraction. We implemented our proposed method on a computer with an Intel Core i5 3.20-GHz processor and a 16 GB RAM.

B. Road Boundary Extraction

We first partitioned the MLS point clouds into a number of blocks using an interval of $100m$ along the trajectory. For each block, specific values were given to the parameters ϵ, θ, R_{seed}, and K in the supervoxel algorithm to segment road points into facets and remove uninterested points. Graph cuts based energy minimization was then used to extract road boundaries. The values of these parameters used in the two datasets are listed in Table II. Specifically, ϵ, θ, R_{seed}, and K are set to the average resolution of original point clouds, 22.5°, 15ϵ and 15, respectively (similar to [35]).

Weighting parameter λ provides a trade-off between the data and the smoothness terms defined in Eq. 11. Specifically, for straight road boundaries, a small λ should be used to favor the data term. In contrast, for curved road boundaries, a large λ should be used to favor the smoothness term. Our boundary extraction results with two different λ values on two different scenes are shown in Fig. 5. Better performance is achieved with a larger value of λ when dealing with crossroads, but a smaller value of λ produces better performance for straight roads surrounded with grass strips. Since λ is a data-related parameter, we selected a training set from each of the two datasets to train λ.

As shown in Figs. 6 and 7, our road boundary extraction results (red points) are superimposed on the MLS point clouds. Several parts of typical road scenes are enlarged for visual inspection. It is observed that road boundaries are accurately extracted by our method. However, since some places have different patterns as compared to their nearest trajectories, the proposed method may fail to extract road boundaries for these places. That is mainly because the trajectory in these places is not consistent with the road boundary. As illustrated in Fig. 8, the #1 and #2 road boundaries were not extracted when the trajectory did not exactly follow the road boundaries.

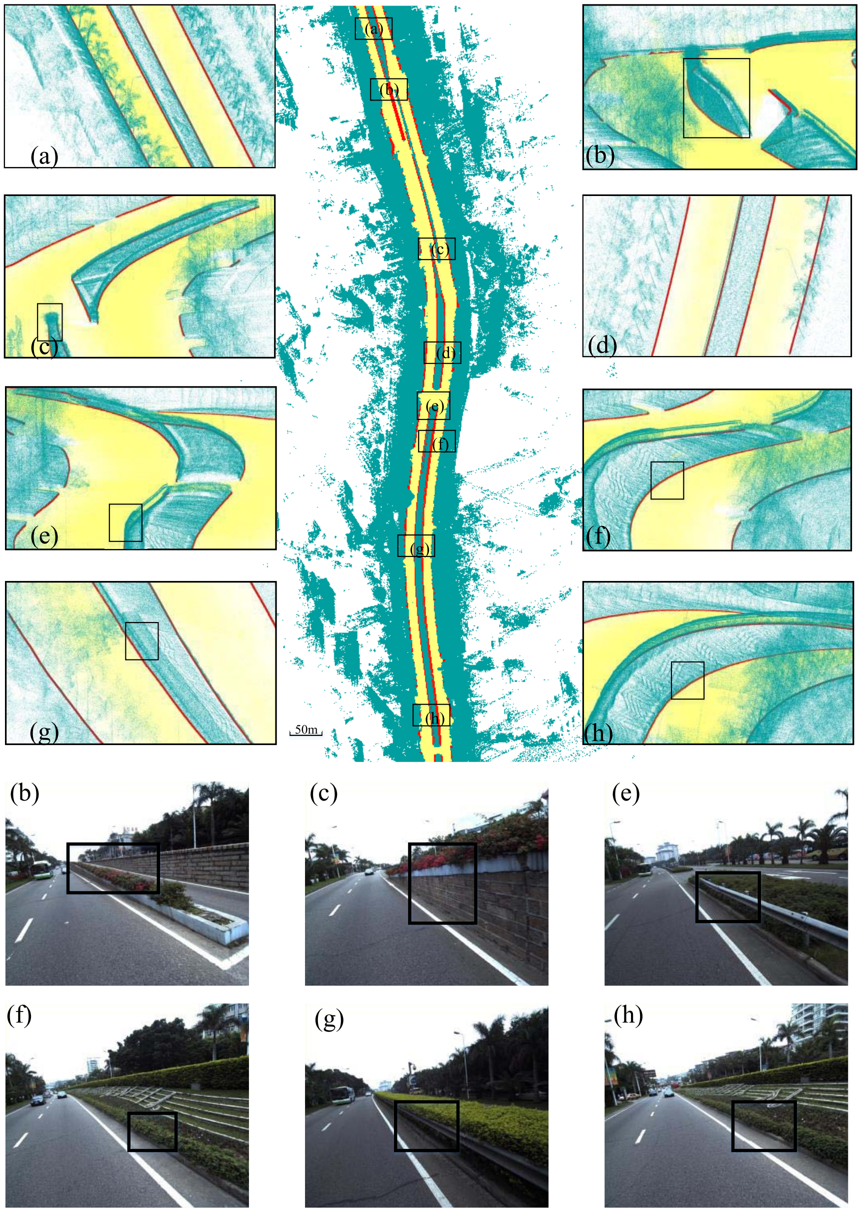


Fig. 7. Road boundary extraction results on the ring road dataset. Irregular road boundaries with a wall, curbs surrounded by fences and tall grass strips, and curbs surrounded by grass strips are shown in black rectangles of Figs. (b, c), (e, g) and (f, h), respectively.

In contrast, if the trajectory has the similar direction as the road boundaries, the road boundaries can be accurately extracted (see #3 boundary in Fig. 8). In practice, several scans should be acquired at different viewpoints to obtain complete point clouds of a scene, which significantly reduces the weakness of our proposed method. As shown in Figs. 6 (d-f), our method accurately extracts road boundaries at crossroads and intersections. Moreover, our method achieves a superior performance even in some highly challenging situations, e.g., irregular road boundaries with a wall (Figs. 7 (b, c)), curbs surrounded by fences and tall grass strips (Figs. 7 (e, g)), and curbs surrounded by grass strips (Figs. 7 (f, h)).

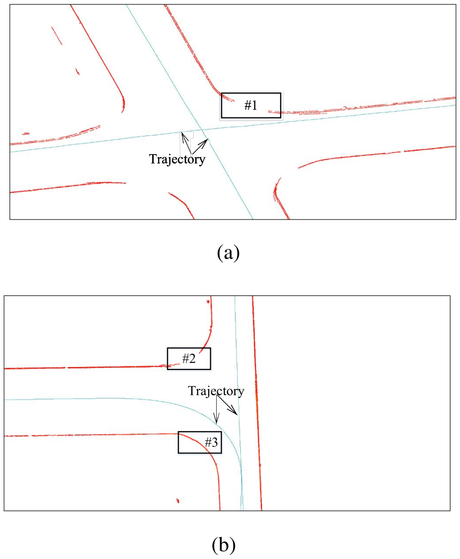
C. Comparative Study

To further demonstrate the performance of our method, we compared it with the methods proposed in [28]–[30]. First, ground-truth road boundary points were extracted manually on the two test datasets. For road boundaries occluded by vehicles or road intersections, the distance between the two endpoints of occluded road boundaries are added into the length of the ground truth road boundary.

To quantitatively test the accuracy of road boundary extraction methods, we used three metrics [28] including completeness, correctness, and quality. These metrics are defined as follows:

$$\begin{cases} \text{Completeness:} & r = \dfrac{TP}{L_r} \\ \text{Correctness:} & p = \dfrac{TP}{L_e} \\ \text{Quality:} & q = \dfrac{TP}{L_e + FN} = \dfrac{TP}{TP + FP + FN} \end{cases} \tag{18}$$

where L_r is the total length of the ground-truth road boundaries, L_e is the total length of the extracted road boundaries,



(a)

(b)

Fig. 8. Road boundary extraction results without curve fitting.

TABLE III
ROAD BOUNDARY EXTRACTION RESULTS

Dataset	Method	L_r(m)	L_e(m)	TP(m)	FP(m)	FN(m)
I	Yang's[28]	9292.79	9053.61	8848.18	205.43	444.61
	Wang's[29]	9292.79	8924.34	8866.31	**58.03**	426.48
	Zai's[30]	9292.79	**9363.65**	**9039.00**	324.65	**253.79**
	Ours	9292.79	9125.16	8984.25	140.91	308.54
II	Yang's[28]	3939.03	3392.29	3323.25	69.04	615.78
	Wang's[29]	3939.03	3575.09	3487.29	87.8	451.74
	Zai's[30]	3939.03	3220.15	3140.43	79.72	798.6
	Ours	3939.03	**3808.04**	**3743.29**	**64.75**	**195.74**

(I is the industrial region dataset; II is the coastal ring road dataset.)

TP is the length of the correctly extracted boundaries, FP is the length of the extracted boundaries that do not exist in the data, and FN is the length of the ground-truth boundaries that are not extracted. Table III shows the meta results used to calculate the evaluation metrics. Table IV shows the final evaluation results. It is clear from Table III that, our method achieves the second best results on the industrial region dataset, which represents a relative simple scene. However, for the more challenging coastal ring road dataset, the proposed method achieves the best performance in all. As shown in Table IV, our method achieves completeness of 96.68% and 95.03%, correctness of 98.46% and 98.30%, and quality of 95.24% and 93.49% on the two datasets, respectively. Our method outperforms the existing methods in terms of Quality on two datasets. Although our method is relatively more time-consuming than existing methods, its running time is still acceptable considering the large number of points (i.e., 331 and 220 million points for the two data sets). Overall, our proposed method is very accurate for road boundary extraction from MLS data.

TABLE IV
QUANTITATIVE EVALUATION RESULTS

Dataset	Method	Completeness (%)	Correctness (%)	Quality (%)	Time (s)
I	Yang's[28]	95.22	97.73	93.16	**4274**
	Wang's[29]	95.41	**99.35**	94.81	4605
	Zai's[30]	**97.27**	96.53	93.99	4820
	Ours	96.6	98.46	**95.24**	5597
II	Yang's[28]	84.37	97.96	82.91	1799
	Wang's[29]	88.53	97.54	86.60	2264
	Zai's[30]	79.73	97.52	78.14	**1454**
	Ours	**95.03**	**98.30**	**93.49**	2897

(I is the industrial region dataset; II is the coastal ring road dataset.)

The good performance of our method is attributed to at least two factors: (1) The improved supervoxel algorithm segments road points into facets and the boundary information can be well preserved. (2) The graph cuts based energy minimization algorithm can effectively extract road boundaries based on the intrinsic characteristics of road boundaries.

V. CONCLUSION

In this paper, we have presented a new method for rapid and automated road boundary extraction from MLS point clouds. Our method was tested on two MLS datasets acquired by a RIEGL VMX-450 system. Our proposed method achieves an average completeness, correctness and quality of more than 95%, 98%, and 94% on two datasets. Comparative experiments clearly demonstrate that our proposed method outperforms the other two methods. It can extract road boundaries more completely and effectively.

REFERENCES

[1] J. McCall and M. M. Trivedi, "Video-based lane estimation and tracking for driver assistance: Survey, system, and evaluation," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 1, pp. 20–37, Mar. 2006.

[2] H. Cheng, N. Zheng, X. Zhang, J. Qin, and H. V. D. 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] C. Rose, J. Britt, J. Allen, and D. Bevly, "An integrated vehicle navigation system utilizing lane-detection and lateral position estimation systems in difficult environments for gps," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 6, pp. 2615–2629, Dec. 2014.

[5] H. Kaartinen, J. Hyyppä, A. Kukko, A. Jaakkola, and H. Hyyppä, "Benchmarking the performance of mobile laser scanning systems using a permanent test field," *Sensors*, vol. 12, no. 9, pp. 12814–12835, 2012.

[6] H. Guan, J. Li, Y. Yu, M. Chapman, and C. Wang, "Automated road information extraction from mobile laser scanning data," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 194–205, Jan. 2015.

[7] 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.

[8] 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.

[9] H. Guan, J. Li, Y. Yu, Z. Ji, and C. Wang, "Using mobile LiDAR data for rapidly updating road markings," *IEEE Trans Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2457–2466, May 2015.

[10] 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, Sep. 2015.

[11] H. Luo *et al.*, "Patch-based semantic labeling of road scene using colorized mobile LiDAR point clouds," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 5, pp. 1286–1297, May 2016.

[12] L. J. Quackenbush, "A review of techniques for extracting linear features from imagery," *Photogramm. Eng. Remote Sens.*, vol. 70, no. 12, pp. 1383–1392, 2004.

[13] J. B. Mena, "State of the art on automatic road extraction for GIS update: A novel classification," *Pattern Recognit. Lett.*, vol. 24, no. 16, pp. 3037–3058, 2003.

[14] J. Wegner, J. Montoya-Zegarra, and K. Schindler, "A higher-order CRF model for road network extraction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Apr. 2013, pp. 1698–1705.

[15] A. Alharthy and J. Bethel, "Automated road extraction from LiDAR data," in *Proc. ASPRS Annu. Conf.*, 2003, pp. 05–09.

[16] Y. W. Choi, Y. W. Jang, H. J. Lee, and G. S. Cho, "Three-dimensional LiDAR data classifying to extract road point in urban area," *IEEE Geosci. Remote Sens. Lett.*, vol. 5, no. 4, pp. 725–729, Oct. 2008.

[17] G. Vosselman and L. Zhou, "Detection of curbstones in airborne laser scanning data," *Int. Archives Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 38, pp. 111–116, Sep. 2009.

[18] L. Zhou and G. Vosselman, "Mapping curbstones in airborne and mobile laser scanning data," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 18, pp. 293–304, Sep. 2012.

[19] A. Boyko and T. Funkhouser, "Extracting roads from dense point clouds in large scale urban environment," *ISPRS J. Photogramm. Remote Sens.*, vol. 66, no. 6, pp. S2–S12, 2011.

[20] X. Hu, Y. Li, J. Shan, J. Zhang, and Y. Zhang, "Road centerline extraction in complex urban scenes from LiDAR data based on multiple features," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 11, pp. 7448–7456, Nov. 2014.

[21] A. Jaakkola, J. Hyyppä, H. Hyyppä, and A. Kukko, "Retrieval algorithms for road surface modelling using laser-based mobile mapping," *Sensors*, vol. 8, no. 9, pp. 5238–5249, 2008.

[22] J. Hernändez and B. Marcotegui, "Filtering of artifacts and pavement segmentation from mobile LiDAR data," in *Proc. ISPRS Workshop Laserscanning*, 2009, pp. 1–6.

[23] P. Kumar, C. P. McElhinney, P. Lewis, and T. McCarthy, "An automated algorithm for extracting road edges from terrestrial mobile LiDAR data," *ISPRS J. Photogramm. Remote Sens.*, vol. 85, pp. 44–55, Nov. 2013.

[24] A. Serna and B. Marcotegui, "Urban accessibility diagnosis from mobile laser scanning data," *ISPRS J. Photogramm. Remote Sens.*, vol. 84, pp. 23–32, Aug. 2013.

[25] L. Smadja, J. Ninot, and T. Gavrilovic, "Road extraction and environment interpretation from LiDAR sensors," in *Proc. IAPRS*, vol. 38. 2010, pp. 281–286.

[26] X. Yuan, C.-X. Zhao, and H.-F. Zhang, "Road detection and corner extraction using high definition LiDAR," *Inf. Technol. J.*, vol. 9, no. 5, pp. 1022–1030, 2010.

[27] S. Ibrahim and D. Lichti, "Curb-based street floor extraction from mobile terrestrial lidar point cloud," *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. 39, pp. 1B–B5, Sep. 2012.

[28] B. Yang, L. Fang, and J. Li, "Semi-automated extraction and delineation of 3D roads of street scene from mobile laser scanning point clouds," *ISPRS J. Photogramm. Remote Sens.*, vol. 79, pp. 80–93, May 2013.

[29] H. Wang *et al.*, "Road boundaries detection based on local normal saliency from mobile laser scanning data," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 10, pp. 2085–2089, Oct. 2015.

[30] D. Zai *et al.*, "3D road surface extraction from mobile laser scanning point clouds," in *Proc. IGARSS*, Jul. 2016, pp. 1595–1598.

[31] R. Lakaemper and L. J. Latecki, "Using extended em to segment planar structures in 3D," in *Proc. IEEE 18th Int. Conf. Pattern Recognit. (ICPR)*, vol. 3. Aug. 2006, pp. 1077–1082.

[32] D. Borrmann, J. Elseberg, K. Lingemann, and A. Nüchter, "The 3D Hough transform for plane detection in point clouds: A review and a new accumulator design," *3D Res.*, vol. 2, no. 2, pp. 1–13, 2011.

[33] R. Schnabel, R. Wahl, and R. Klein, "Efficient RANSAC for point-cloud shape detection," *Comput. Graph. Forum*, vol. 26, no. 2, pp. 214–226, 2007.

[34] A.-L. Chauve, P. Labatut, and J.-P. Pons, "Robust piecewise-planar 3D reconstruction and completion from large-scale unstructured point data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 1261–1268.

[35] Y. Lin, C. Wang, B. Chen, D. Zai, and J. Li, "Facet segmentation based line segment extraction for large-scale point clouds," *IEEE Trans. Geosci. Remote Sens.*, doi:10.1109/TGRS.2016.2639025.

[36] J. Papon, A. Abramov, M. Schoeler, and F. Worgotter, "Voxel cloud connectivity segmentation—Supervoxels for point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2027–2034.

[37] A. M. Ramiya, R. R. Nidamanuri, and K. Ramakrishnan, "A supervoxel-based spectro-spatial approach for 3D urban point cloud labelling," *Int. J. Remote Sens.*, vol. 37, no. 17, pp. 4172–4200, 2016.

[38] P. Babahajiani, L. Fan, J. Kamarainen, and M. Gabbouj, "Automated super-voxel based features classification of urban environments by integrating 3D point cloud and image content," in *Proc. IEEE Int. Conf. Signal Image Process. Appl. (ICSIPA)*, Oct. 2015, pp. 372–377.

[39] S. Song, H. Lee, and S. Jo, "Boundary-enhanced supervoxel segmentation for sparse outdoor LiDAR data," *Electron. Lett.*, vol. 50, no. 25, pp. 1917–1919, 2014.

[40] Y. Guo, M. Bennamoun, F. Sohel, and J. Wan, "3D object recognition in cluttered scenes with local surface features: A survey," *IEEE Trans Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2270–2287, Nov. 2014.

[41] H. Edelsbrunner, D. G. Kirkpatrick, and R. Seidel, "On the shape of a set of points in the plane," *IEEE Trans. Inf. Theory*, vol. 29, no. 4, pp. 551–559, Jul. 1983.

[42] Y. Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images," in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 1. Sep. 2001, pp. 105–112.

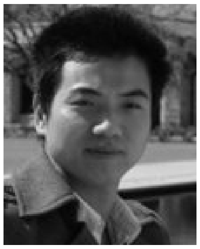
[43] V. Kolmogorov and R. Zabin, "What energy functions can be minimized via graph cuts?" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 147–159, Feb. 2004.

[44] I. Stainvas and Y. Buda, "Performance evaluation for curb detection problem," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2014, pp. 25–30.

[45] U. Ozertem and D. Erdogmus, "Locally defined principal curves and surfaces," *J. Mach. Learn. Res.*, vol. 12, pp. 1249–1286, Feb. 2011.

Dawei Zai received the B.S. degree in aircraft design and engineering from Xian Jiaotong University, Xi'an, China, in 2010. He is currently working toward the Ph.D. degree with the Fujian Key Laboratory of Sensing and Computing for Smart City, School of Information Science and Engineering, Xiamen University, China. His research interests include 3-D vision, and machine learning.

Jonathan Li (M'00–SM'11) received the Ph.D. degree in geomatics engineering from University of Cape Town, South Africa. He is currently a Professor and the Head of the Mobile Sensing and Geodata Science Laboratory, Department of Geography and Environmental Management, University of Waterloo, Canada. He has co-authored over 350 publications, over 150 of which were published in refereed journals, including IEEE-TGRS, IEEETITS, IEEE-GRSL, IEEEJSTARS, ISPRS-JPRS, IJRS, PE&RS, and RSE. His research interests include information extraction from LiDAR point clouds and from earth observation images. He will be the Chair of the ISPRS Working Group I/2 on LiDAR for Airborne and Spaceborne Sensing, from 2016 to 2020, the Chair of the ICA Commission on Sensor driven Mapping, from 2015 to 2019, and an Associate Editor of IEEE-TITS and IEEEJSTARS.



Yulan Guo received the B.Eng. and Ph.D. degrees in information and communication engineering from National University of Defense Technology (NUDT), Changsha, China, in 2008 and 2015, respectively. From 2011 to 2014, he was a Visiting Ph.D. Student with The University of Western Australia. He is currently an Assistant Professor with the College of Electronic Science and Engineering, NUDT. He has authored over 30 peer-reviewed journal and conference publications, including IEEE TPAMI and IJCV. He served as a Reviewer of over 30 international journals and conferences. His research interests include computer vision, and pattern recognition, particularly on 3-D feature extraction, 3-D modeling, 3-D object recognition, and 3-D face recognition. He was a recipient of the NUDT Distinguished Ph.D. Thesis Award in 2015 and the CAAI Distinguished Ph.D. Thesis Award in 2016.

Ming Cheng received the Ph.D. degree in biomedical engineering from Tsinghua University, China, in 2004. He is currently an Associate Professor with Fujian Key Laboratory of Sensing and Computing for Smart Cities and the Xiamen Key Laboratory of Geospatial Sensing and Computing, School of Information Science and Engineering, Xiamen University, Xiamen, China. He has authored over 30 papers in refereed journals and conference proceedings, including IEEE GEOSCIENCE AND REMOTE SENSING LETTERS, *Neurocomputing*, and the IGARSS and ISPRS Proceedings. His research interests include remote sensing image processing, point cloud processing, computer vision, and machine learning.

Yangbin Lin received the B.Sc., M.Sc., and Ph.D. degrees in computer science from Xiamen University, Xiamen, China, in 2008, 2011, and 2016, respectively. He was a Research Assistant with University of Hong Kong in 2010. He was a Software Engineer with Google Company from 2011 to 2012. He is currently a new Faculty Member with the Computer Engineering College, Jimei University, Xiamen. His research interests include point cloud, graphics, and optimization.

Huan Luo received the B.Sc. degree in software engineering from Nanchang University, Nanchang, China, in 2009. He is currently working toward the Ph.D. degree with Fujian Key Laboratory of Sensing and Computing for Smart City, School of Information Science and Engineering, Xiamen University, China. His research interests include computer vision, machine learning, and mobile LiDAR point cloud data processing.

Cheng Wang received the Ph.D. degree in information communication engineering from University of Defense Technology, Changsha, China, in 2002. He is currently a Professor and an Associate Dean of the School of Information Science and Engineering, Xiamen University, China, where he is also the Executive Director of Fujian Key Laboratory of Sensing and Computing for Smart City. He has co-authored over 80 papers in refereed journals, including IEEE-TGRS, IEEE-TITS, IEEE-GRSL, IEEE-JSTARS, IJRS, and ISPRS-JPRS. His research interests include remote sensing image processing, mobile LiDAR data analysis, and multisensory fusion. He is a Co-Chair of the ISPRS WG I/3 on Multi-Platform Multi-Sensor System Calibration, from 2012 to 2016, and a Council Member of the China Society of Image and Graphics.