

# Road Curb Extraction From Mobile LiDAR Point Clouds

Sheng Xu, Ruisheng Wang, and Han Zheng

**Abstract**—Automatic extraction of road curbs from uneven, unorganized, noisy, and massive 3-D point clouds is a challenging task. Existing methods often project 3-D point clouds onto 2-D planes to extract curbs. However, the projection causes loss of 3-D information, which degrades the performance of the detection. This paper presents a robust, accurate, and efficient method to extract road curbs from 3-D mobile LiDAR point clouds. Our method consists of two steps: 1) extracting candidate points of curbs based on the proposed novel energy function and 2) refining candidate points using the proposed least cost path model. We evaluated the method on a large scale of residential area (16.7 GB, 300 million points) and an urban area (1.07 GB, 20 million points) mobile LiDAR point clouds. Results indicate that the proposed method is superior to the state-of-the-art methods in terms of robustness, accuracy, and efficiency. The proposed curb extraction method achieved a completeness of 78.62% and a correctness of 83.29%. Experiments demonstrate that our method is a promising solution to extract road curbs from mobile LiDAR point clouds.

**Index Terms**—3-D point clouds, energy function, least cost path, mobile LiDAR, road curbs.

## I. INTRODUCTION

MOBILE LiDAR systems (MLS) are a newly emerging technology which collects 3-D information of objects while vehicles drive at a posted speed [1]. It becomes more and more popular in analyzing 3-D point clouds because of its high density, efficiency, and cost-effectiveness and provides the possibility to extract the microobjects, such as road curbs.

Road curb extraction from 3-D point clouds is a basis for several types of research, such as road surface analysis, driving simulation, safe parking, autonomous driving, and traffic environment understanding. However, point clouds acquired by MLS are often found to be uneven, unorganized, noisy, and massive, thereby making the curb detection a challenging task.

In this paper, we present a robust, accurate, and efficient method to extract road curbs from mobile LiDAR point clouds. The main contributions of this paper are the following: 1) we propose a novel energy function to extract candidate points of curbs from mobile LiDAR point clouds; 2) we propose

a least cost path model (LCPM) to link candidate points into complete curbs; and 3) we conduct a comprehensive evaluation of the proposed method using large-scale data.

This paper is organized as follows. In Section II, we review the state-of-the-art methods related to road curb extraction. In Section III, we propose a new energy function to extract candidate points of curbs and design an LCPM to connect candidate points into global optimization curbs. In Section IV, we evaluate the robustness, accuracy, and complexity of the proposed method. The conclusions are presented in Section V.

## II. RELATED WORK

A straightforward method for road curb detection usually makes use of elevation information. For example, algorithms [2]–[6] focus on detecting objects in terms of the elevation difference. It is possible to obtain road curbs by elevation filtering, however, at the compromise of robustness. There are no reliable cues to design adaptive thresholds for the low elevation curbs in different scenes. These methods produce attractive results in the straight roads with the same elevation. Nonetheless, they fail to work in occluded, sunken, or uphill road areas.

Recent methods [7]–[10] focus on models that incorporate more prior knowledge, such as width, elevation, and density, to form descriptors for classifying regions and edges. The prior knowledge-based method [11] uses a predefined curb model in terms of the elevation jump, point density, and slope change to provide potential location of curbs from mobile LiDAR point clouds. However, this method does not work well on data sets with different geometrical features, such as large slopes or uneven road surface, due to the use of nonrobust 3-D features.

The typical technique for boundary extraction is the active contour model (Snake) [12]. Snake is widely applied to curb extraction from images or images generated from 3-D point clouds' projection [13]–[15]. However, the projection loses 3-D information that will degrade the performance of the extraction. This is also the main drawback of the extraction methods [16]–[18] based on 2-D images. Moreover, Snake needs a manual initialization to start the iteration.

The methods [19], [20] combine LiDAR point clouds and the corresponding images to detect road curbs, but fail to work when there are occlusions caused by cars, pedestrians, or trees along the road. Moreover, the registration of LiDAR point clouds and images is not reliable due to the duplication of moving objects.

Manuscript received April 4, 2016; revised June 8, 2016 and September 7, 2016; accepted October 4, 2016. Date of publication October 26, 2016; date of current version December 29, 2016.

The authors are with the Department of Geomatics Engineering, University of Calgary, Calgary, AB T2N 1N4, Canada (e-mail: sheng.xu2@ucalgary.ca; ruishwang@ucalgary.ca; han.zheng@ucalgary.ca).

This paper has supplementary downloadable material available at <http://ieeexplore.ieee.org>, provided by the author.

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/TGRS.2016.2617819

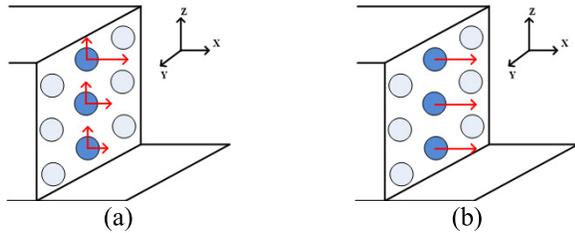


Fig. 1. Different definitions of the 3-D gradient. (a) Elevation difference. (b) Intensity difference.

In summary, the challenges in curb detection from mobile LiDAR point clouds are as follows. First, the data are difficult to process due to its uneven, unorganized, noisy, and massive nature. Second, there is no much reliable information, such as color, intensity, and texture, for segmentation or classification. Third, since the LiDAR sensors mounted on the moving vehicle scan objects line by line, there will be misalignments or duplicates of moving objects and occlusions.

This paper aims to provide a robust solution for road curb extraction from mobile LiDAR point clouds. In a pre-processing step, we remove the nonground points by using elevation histogram and organize the ground point clouds into voxels. Then, we extract candidate points of curbs using the proposed energy function. Finally, we use the proposed LCPM to complete optimal curbs.

Comparison with the existing curb extraction algorithms, our method has no risk of losing 3-D information. Since all existing methods lack experiments on large-scale data, we evaluate our algorithm on large-scale residential and medium-sized urban data to verify the proposed method.

### III. METHOD

#### A. Definition of 3-D Sampling Density and Density Gradient

In 2-D images, the gradient shows the increase or decrease in the magnitude of the intensity. However, to the best of our knowledge, no unified definition of the gradient exists for 3-D point clouds to date. This paper analogizes the gradient in 2-D to obtain the gradient definition in 3-D, which only uses the geometric information of the MLS measurement.

In this paper, the gradient concept is extended to 3-D point clouds through considering the points' density in a local area. At first, voxels are generated for the point cloud. Then, the intensity of each voxel is defined by the points' density, i.e., the number of the point inside the voxel. Finally, our 3-D sampling density gradient is calculated by the difference of the intensity between adjacent voxels. The intensity is approximated by the number of points in a local area.

One existing 3-D gradient definition is based on elevation difference between adjacent points. In Fig. 1(a), the elevation along  $z$ -axis is increasing evenly, so the gradient is a constant along  $z$ -axis and zero along  $y$ -axis. However, the gradient along  $x$ -axis varies due to different elevations. This is not desirable, because the gradients along the normal direction of the façade (i.e., along  $x$ -axis) are different, as shown in Fig. 1(a). Our definition of 3-D sampling density gradient

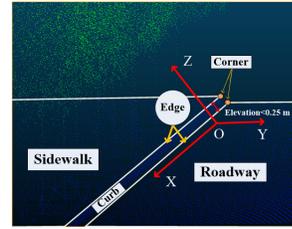


Fig. 2. Three areas in road point clouds, namely, sidewalk, roadway, and curb.

is based on the intensity difference between the adjacent points. Our gradient is zero along either  $y$ -axis or  $z$ -axis and a constant along  $x$ -axis, as shown in Fig. 1(b), which better represents the real situation. We use this new 3-D intensity and sampling density gradient definition throughout this paper.

Similar to the change produced by a shift for a pixel in the 2-D image [21], we use (1)

$$C(x, y, z) = \sum_{\Delta x, \Delta y, \Delta z} [I_{\Delta x+x, \Delta y+y, \Delta z+z} - I_{x, y, z}]^2 \quad (1)$$

to define the magnitude of change (squared) in the diagonal direction for a point  $p(x, y, z)$  in a small shift  $(\Delta x, \Delta y, \Delta z)$  where  $I_{x, y, z}$  is the intensity, which is the number of points in a local area around  $p$ . Our point clouds are uneven and unorganized. The Euclidean distance between each point is various and neighbors of each point are unknown. Thus, we use voxel of a suitable size to represent each local area and organize 3-D point clouds in a sparse 3-D matrix. The value of each voxel is the intensity, which is the number of points in each voxel. Finally, we can deal our data, such as pixels, in 2-D image with the voxel-based representation.  $(\Delta x, \Delta y, \Delta z)$  means the coordinate difference between two voxels, which can also be treated as a directional vector, for example,  $(1, 0, 0)$  is the direction of the  $x$ -axis,  $(1, 1, 0)$  is the direction of  $45^\circ$  in the  $XOY$ ,  $(1, 0, 1)$  is the direction of  $45^\circ$  in the  $XOZ$ , and so on. The coordinate difference should be an integer.

#### B. Classification of the Road Areas

There are mainly three regions in road point clouds, as shown in Fig. 2: roadway, sidewalk, and curb. The curb connects the roadway and the sidewalk, and it is usually lower than 0.25 m. The Euclidean distance between two points in our data is larger than 0.004 m. The elevation difference between sidewalk and roadway is small. Thus, the road curb detection cannot heavily rely on the elevation difference.

Suppose that the point clouds are aligned with a 3-D coordinate system  $O\text{-}XYZ$ , as shown in Fig. 2, the curb is in parallel with the  $x$ -axis and  $xOz$  plane, and roadway and sidewalk are in parallel with the  $xOy$  plane.

Denote  $G_x$ ,  $G_y$ , and  $G_z$  as our sampling density gradients of a voxel along the  $x$ -axis,  $y$ -axis, and  $z$ -axis directions, respectively, as shown in Fig. 3. There will be three primary situations for our gradients of a voxel in Fig. 2.

- 1) The voxel within the surface: there is only one large gradient, such as the large  $G_y$  in the curb areas and the large  $G_z$  in the roadway or sidewalk areas.

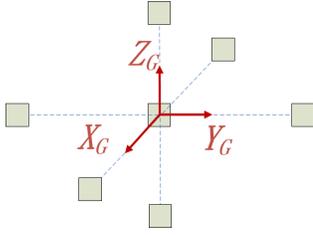


Fig. 3. Sampling density gradients of a voxel in each axis direction.

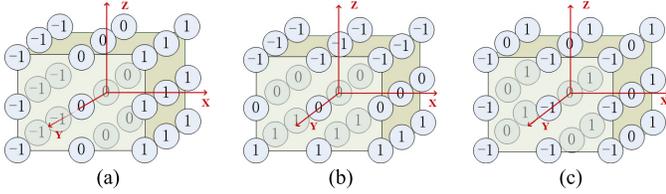


Fig. 4. Three-dimensional Sobel. (a)  $\text{Cube}_x$ . (b)  $\text{Cube}_y$ . (c)  $\text{Cube}_z$ .

- 2) The voxel in the intersection of two surfaces: there are more than one large gradient, such as the large  $G_y$  and  $G_z$  along the curb edges.
- 3) The voxel in the intersection of three mutually nonparallel surfaces: all gradients are large, such as the curb corners.

Therefore, we conclude that if gradients of a voxel are large in more than one direction, the points in this voxel potentially belong to the curb.

### C. Mathematical Model

Equation (1) can be rewritten as (2), the Taylor series expansion with  $O(\Delta x^2, \Delta y^2, \Delta z^2)$  as the remainder term

$$C(x, y, z) = \sum_{\Delta x, \Delta y, \Delta z} [\Delta x \cdot X_G + \Delta y \cdot Y_G + \Delta z \cdot Z_G + O(\Delta x^2, \Delta y^2, \Delta z^2)]^2 \quad (2)$$

where  $X_G$ ,  $Y_G$ , and  $Z_G$  are the sampling density gradients, which are defined as

$$\begin{aligned} X_G &= \left( \frac{\partial I}{\partial x} \right) \approx I \cdot \text{Cube}_x \\ Y_G &= \left( \frac{\partial I}{\partial y} \right) \approx I \cdot \text{Cube}_y \\ Z_G &= \left( \frac{\partial I}{\partial z} \right) \approx I \cdot \text{Cube}_z \end{aligned} \quad (3)$$

where  $\text{Cube}_x$ ,  $\text{Cube}_y$ , and  $\text{Cube}_z$  are three  $3 \times 3 \times 3$  operators extended from Sobel [22], as shown in Fig. 4.  $I$  is a  $3 \times 3 \times 3$  matrix whose elements are the intensity of each voxel and its neighbors.

We obtain  $C(x, y, z)$  for each voxel as

$$\begin{aligned} C(x, y, z) &= (X_G \cdot \Delta x)^2 + (Y_G \cdot \Delta y)^2 + (Z_G \cdot \Delta z)^2 + 2 \\ &\quad \times \Delta x \cdot \Delta y \cdot X_G \cdot Y_G + 2 \times \Delta y \cdot \Delta z \cdot Y_G \cdot Z_G \\ &\quad + 2 \times \Delta x \cdot \Delta z \cdot X_G \cdot Z_G \\ &= (\Delta x, \Delta y, \Delta z) \mathbf{M} (\Delta x, \Delta y, \Delta z)^T \geq 0 \end{aligned} \quad (4)$$

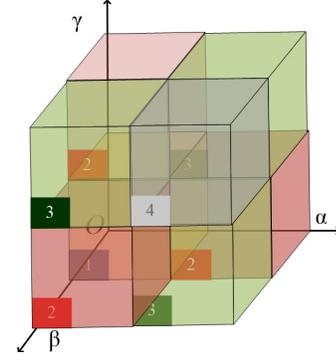


Fig. 5. Four areas for different ranges of the eigenvalues  $\alpha$ ,  $\beta$ , and  $\gamma$ . 1: All eigenvalues are small. 2: Only one large eigenvalue. 3: Two large eigenvalues. 4: All eigenvalues are large.

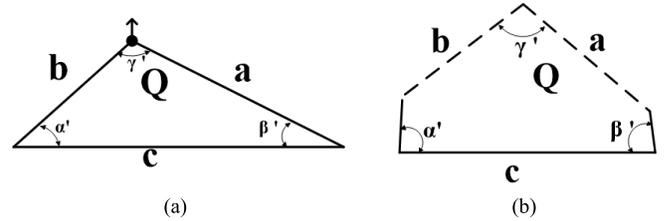


Fig. 6. Related triangle. (a) Initial triangle  $Q$ . (b) Increase the height of  $Q$ .

where  $\mathbf{M}$  is

$$\begin{pmatrix} X_G^2 & X_G \cdot Y_G & X_G \cdot Z_G \\ X_G \cdot Y_G & Y_G^2 & Y_G \cdot Z_G \\ X_G \cdot Z_G & Y_G \cdot Z_G & Z_G^2 \end{pmatrix}.$$

$\mathbf{M}$  is a semipositive symmetric matrix. Its eigenvectors are mutually orthogonal, and eigenvalues  $\alpha$ ,  $\beta$ , and  $\gamma$  are not less than 0. According to road areas, three possibilities will be observed for each voxel: 1) the voxel belongs to the surface area, when only one large eigenvalue exists; 2) the voxel belongs to two surfaces' intersection when two large eigenvalues are observed; and 3) the voxel belongs to the intersection area of three mutually nonparallel surfaces, when three large eigenvalues are observed.

In 2-D images, the edges have a large gradient in one direction. The intersection of two edges is the corners, which have a large gradient in more than one direction. Similarly, for our sampling density gradient in 3-D point clouds, a large gradient in one direction means planar surface areas. The edges are the intersection of two surfaces. Thus, the candidate points of curb edges have large gradients in at least two directions. Now, our sampling density gradients are related to the eigenvalues of the matrix  $\mathbf{M}$ . We show different ranges of eigenvalues in Fig. 5. There are four areas, such as all of eigenvalues are small (area 1), only one of eigenvalues is large (area 2), two of eigenvalues are large (area 3), and all of eigenvalues are large (area 4).

We can decide the area of each voxel by a thresholding method. However, it is difficult to tune different thresholds for each eigenvalue. Thus, we form an energy function based on the eigenvalues  $\alpha$ ,  $\beta$ , and  $\gamma$ . The energy corresponding to each voxel can map to the areas in Fig. 5. If the result of mapping is in area 3 or area 4, the voxel is chosen as the candidate curb edges.

### D. Construction and Analysis of the Energy Function

We design the energy function based on a triangle  $Q$ . The energy will be related to the area  $S$  of this triangle. The eigenvalues  $\alpha$ ,  $\beta$ , and  $\gamma$  are related to angles  $\alpha'$ ,  $\beta'$ , and  $\gamma'$  of  $Q$ . Sides  $a$ ,  $b$ , and  $c$  are in opposite to  $\alpha'$ ,  $\beta'$ , and  $\gamma'$ , respectively, as shown in Fig. 6. Sampling density gradients large in two or three directions are now related to the triangle  $Q$  with at least two large angles.

Fig. 6(a) is the initial triangle. Increase the height of the triangle  $Q$  to change the base angles  $\alpha'$  and  $\beta'$  together. One can find that the area  $S$  is growing as increasing the two base angles. If the angle  $\gamma'$  is infinitely close to 0,  $S$  achieves the maximum, as shown in Fig. 6(b).

Our idea is to relate the energy  $E$  to the area  $S$  to ensure that, if  $E$  is large, there will be more than one large eigenvalue. The challenges are relations among angles, eigenvalues, and the length of sides. The following is the derivation of the energy function.

Let

$$\begin{aligned} \varphi &= \frac{\pi}{\arctan \alpha + \arctan \beta + \arctan \gamma} \\ \alpha' &= \arctan \alpha \times \varphi, \quad \gamma' = \arctan \gamma \times \varphi, \quad \beta' = \arctan \beta \times \varphi. \end{aligned} \quad (5)$$

The area  $S$  is derived according to the sine theorem

$$\begin{aligned} S &= \frac{1}{2} a \times b \sin \gamma' = \frac{1}{2} \times \frac{\sin \beta'}{\sin \gamma'} \times \frac{\sin \alpha'}{\sin \gamma'} \times \sin \gamma' \times c^2 \\ &= \frac{c^2}{2} \times \frac{\sin \alpha' \times \sin \beta'}{\sin(\pi - (\alpha' + \beta'))} \\ &= \frac{c^2}{2} \times \frac{\sin \alpha' \times \sin \beta'}{\sin \alpha' \cos \beta' + \cos \alpha' \sin \beta'} \\ &= \frac{c^2}{2} \times \frac{\tan \alpha' \times \tan \beta'}{\tan \alpha' + \tan \beta'} \end{aligned} \quad (6)$$

and

$$\begin{aligned} \tan \alpha' &= \tan(\arctan \alpha \times T) \\ &= \tan \left( \frac{\arctan \alpha}{\arctan \alpha + \arctan \beta + \arctan \gamma} \times \pi \right) \\ \tan \beta' &= \tan(\arctan \beta \times T) \\ &= \tan \left( \frac{\arctan \beta}{\arctan \alpha + \arctan \beta + \arctan \gamma} \times \pi \right). \end{aligned} \quad (7)$$

Both  $\tan$  and  $\arctan$  are monotonically increasing functions, so we have

$$\begin{aligned} S &\propto \tan \alpha' \propto \arctan \alpha \propto \alpha \\ S &\propto \tan \beta' \propto \arctan \beta \propto \beta. \end{aligned} \quad (8)$$

Relating the energy  $E$  to the area  $S$ , we have

$$E \propto \frac{\alpha\beta}{\alpha + \beta} \times c^2. \quad (9)$$

The side  $c$  can be regarded as the weighting coefficient. To visualize the energy  $E$ , we calculate the result of (9) using a range  $[0, 10000]$  for  $\alpha$  and  $\beta$ . We can define  $c$  as a constant 1 or a variable. The sum of the eigenvalue  $\alpha$  and

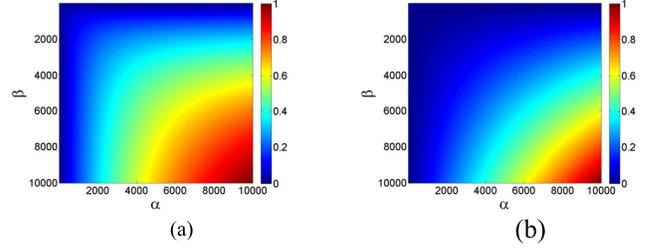


Fig. 7. Visualization of the energy  $E$ . (a)  $c = 1$ . (b)  $c = (\alpha + \beta)$ .

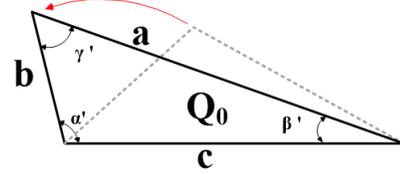


Fig. 8. Increase the area  $S$  by enlarging only one base angle  $\alpha'$ .

$\beta$  is a real number, thus we can define  $c$  as  $(\alpha + \beta)$ . Under this definition, the larger the  $\alpha$  and  $\beta$  values, the larger the  $E$  value. Fig. 7(a) and (b) is the visualization of  $E$  when  $c$  is a constant 1 and a variable  $(\alpha + \beta)$ , respectively. As seen from Fig. 7, when  $c$  is defined as  $(\alpha + \beta)$ , large energy corresponds to the situation where both  $\alpha$  and  $\beta$  are large.

With increasing the height of the triangle  $Q$  by enlarging only one base angle  $\alpha'$ , as shown in Fig. 8, the area  $S$  is also increased. It means that large energy  $E$  can be observed in noncurb areas with only one large eigenvalue. To address this problem, we refined (9).

In the obtuse triangle  $Q_0$ , the angles  $\beta'$  and  $\gamma'$  are small. If we treat the side  $a$  as the fixed bottom side and do the same analysis as in  $Q$ , its area  $S$  will be smaller than the triangle with a large  $\beta'$  and  $\gamma'$ . Thus, we calculate the sum of areas based on all three sides (10) to ensure that the sum  $S'$  is large only when there is more than one large angle, and consequently, we have

$$\begin{aligned} S' &= \frac{c^2}{2} \times \frac{\tan \alpha' \times \tan \beta'}{\tan \alpha' + \tan \beta'} + \frac{a^2}{2} \times \frac{\tan \beta' \times \tan \gamma'}{\tan \beta' + \tan \gamma'} \\ &\quad + \frac{b^2}{2} \times \frac{\tan \alpha' \times \tan \gamma'}{\tan \alpha' + \tan \gamma'}. \end{aligned} \quad (10)$$

For our energy function in 3-D space,  $c$  is unified as  $(\alpha + \beta + \gamma)$  and we relate our energy to the sum  $S'$  as

$$\begin{aligned} E &\propto \frac{\alpha\beta}{\alpha + \beta} \times (\alpha + \beta + \gamma)^2 + \frac{\alpha\gamma}{\alpha + \gamma} \times (\alpha + \beta + \gamma)^2 \\ &\quad + \frac{\gamma\beta}{\gamma + \beta} \times (\alpha + \beta + \gamma)^2 \end{aligned} \quad (11)$$

$(\alpha + \beta + \gamma)^2$  is regarded as the coefficient and we let

$$E = \left( \frac{\alpha\beta}{\alpha + \beta} + \frac{\alpha\gamma}{\alpha + \gamma} + \frac{\gamma\beta}{\gamma + \beta} \right) \times (\alpha + \beta + \gamma)^2. \quad (12)$$

We plot the energy  $E$  based on different ranges of the eigenvalues  $\alpha$ ,  $\beta$ , and  $\gamma$  using (12), as shown in Fig. 9. The

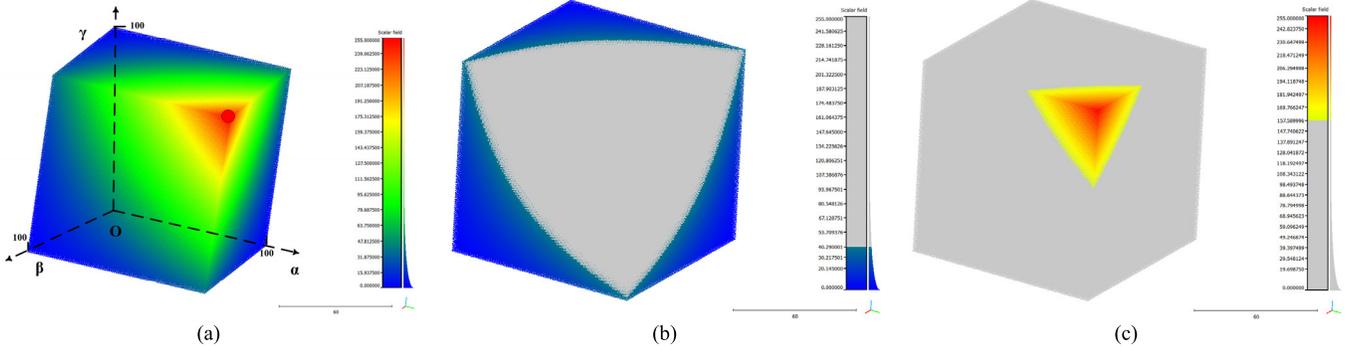


Fig. 9. Three-dimensional visualization of the energy function  $E$ . (a) Plot of  $E$  based on different ranges of  $\alpha$ ,  $\beta$ , and  $\gamma$ . (b)  $E$  is small ( $<40$ ), which corresponds to areas 1 and 2 in Fig. 5. (c)  $E$  is large ( $>157$ ), which corresponds to areas 3 and 4 in Fig. 5.

magnitude of  $E$  is scaled to  $[0, 255]$ , and  $\alpha$ ,  $\beta$ , and  $\gamma$  are ranged from 1 to 100. From Fig. 9(a), any two of  $\alpha$ ,  $\beta$ , and  $\gamma$  are large results in a large  $E$  and the largest  $\alpha$ ,  $\beta$ , and  $\gamma$  causes the maximum  $E$  indicated by the red corner. If the voxel is in the block or surface,  $E$  is small ( $<40$ ), as shown in Fig. 9(b), which corresponds to areas 1 and 2 in Fig. 5. If the voxel is in the curb edges or corners,  $E$  is large ( $>157$ ), as shown in Fig. 9(c), which corresponds to areas 3 and 4 in Fig. 5. The other  $E$  (2616510 [40, 157]) is the unreliable areas correspond to area 3 in Fig. 5. This energy function will be used to detect curbs from road points.

In this paper, voxels corresponding to the top 20% energy are chosen as candidate curbs. Practically, we obtain the energy  $E$  without the computation of eigenvalues or eigenvectors of the matrix  $\mathbf{M}$ . To calculate  $E$  in a low complexity, we decompose the matrix  $\mathbf{M}$  as  $\mathbf{M}_{xoy}$ ,  $\mathbf{M}_{xoz}$ , and  $\mathbf{M}_{yoz}$  on the  $XOY$ ,  $YOZ$ , and  $xoz$  planes, respectively, as follows:

$$\begin{aligned}\mathbf{M}_{xoy} &= \begin{pmatrix} X_G \cdot X_G & X_G \cdot Y_G \\ X_G \cdot Y_G & Y_G \cdot Y_G \end{pmatrix} \\ \mathbf{M}_{xoz} &= \begin{pmatrix} X_G \cdot X_G & X_G \cdot Z_G \\ X_G \cdot Z_G & Z_G \cdot Z_G \end{pmatrix} \\ \mathbf{M}_{yoz} &= \begin{pmatrix} Y_G \cdot Y_G & Y_G \cdot Z_G \\ Y_G \cdot Z_G & Z_G \cdot Z_G \end{pmatrix}.\end{aligned}$$

Now (12) can be calculated effectively by

$$\begin{aligned}E &= \frac{Det(\mathbf{M}_{xoy})}{Tr(\mathbf{M}_{xoy})} \times Tr(\mathbf{M})^2 + \frac{Det(\mathbf{M}_{xoz})}{Tr(\mathbf{M}_{xoz})} \times Tr(\mathbf{M})^2 \\ &\quad + \frac{Det(\mathbf{M}_{yoz})}{Tr(\mathbf{M}_{yoz})} \times Tr(\mathbf{M})^2\end{aligned}\quad (13)$$

where  $Det$  means the determinant and  $Tr$  is the trace. From (13), there is no need to compute the eigenvalues or eigenvectors of the matrix. The complexity of the energy computation is linear time, which is significant for large-scale point cloud processing.

It is worth pointing out that  $Det(\mathbf{M}_{xoy})$ ,  $Det(\mathbf{M}_{xoz})$ , and  $Det(\mathbf{M}_{yoz})$  are 0, because we use the discrete form (3) to obtain the sampling density gradient approximately. To overcome this problem, we smooth the input  $I$  by convolving a  $3 \times 3 \times 3$

Gaussian kernel  $h$

$$h(i, j, k) = \frac{1}{\sqrt{2\pi}\sigma^2} e^{-\frac{(i-1)^2+(j-1)^2+(k-1)^2}{2\sigma^2}}\quad (14)$$

where  $\sigma$  is the standard deviation of the voxels in the convolutional operation. For example, the voxel at  $(x, y, z)$  is smoothed as

$$\begin{aligned}I_h(x, y, z) &= I(x, y, z) \otimes h \\ &= \sum_{i=0}^2 \sum_{j=0}^2 \sum_{k=0}^2 (I(x, y, z) \cdot h(i, j, k))\end{aligned}\quad (15)$$

where the output  $I_h$  is the smoothed input  $I$ . From the associative property of convolution for linear system, the gradient of the convolution is equal to the convolution of the gradient, as shown in (16). Thus, the gradient calculated from  $I_h$  is also smoothed and  $Det(\mathbf{M}_{xoy})$ ,  $Det(\mathbf{M}_{xoz})$ , and  $Det(\mathbf{M}_{yoz})$  are not 0

$$\frac{\partial(I(x, y, z) \otimes h)}{\partial x} = \frac{\partial I(x, y, z)}{\partial x} \otimes h.\quad (16)$$

### E. Least Cost Path Model

Candidate curb edges obtained by optimizing the energy function are incomplete and noisy. We can refine the curbs by line fitting methods, such as least square (LS) fitting [23], Hough transform (HT) [24], and random sample consensus (RANSAC) [25]. Nevertheless, these methods highly rely on the number of candidate curb points and do not consider the noncandidate points and the cost for linking candidate points. The linking of the curb candidate points with few inliers is problematic. We propose a new robust method to link curb candidate points into a complete curb.

Our model consists of a data term and a smoothness term, as shown in (17), to represent different refinement paths.  $N$  is the number of nodes in the final path,  $(u, v, w)$  is the coordinate of the current node  $i$ , and  $j$  is the node prior to  $i$  in the path.  $L_N$  is the cost of the refinement path

$$L_N = \sum_i^N (\text{Data}_i(u, v, w) + \text{Smoothness}_{i,j}(u, v, w)).\quad (17)$$

The data term refers to the cost of a path containing all selected nodes and the smoothness term is the cost connecting

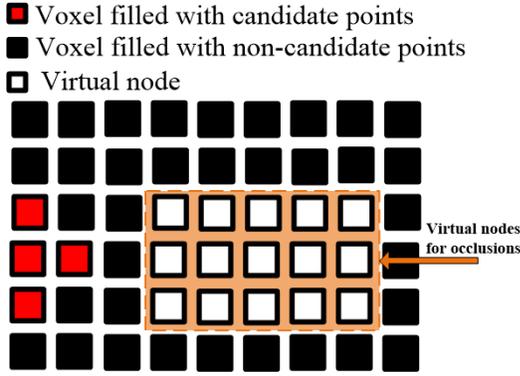


Fig. 10. Nodes formed by voxels.

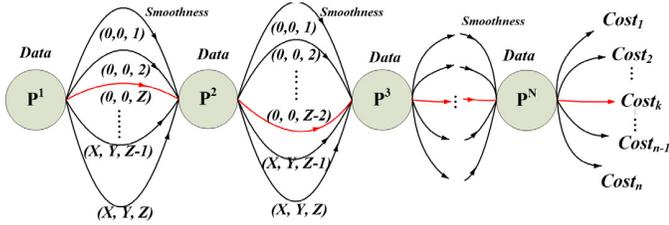


Fig. 11. Graph to be optimized by the LCPM. This graph consists of  $N$  nodes, namely,  $P^1, P^2, P^3, \dots, P^N$ . The cost of each node and edge is calculated by the data term and smoothness term. There will be  $(X \times Y \times Z) \times (N - 1)$  edges in the graph leading to  $(X \times Y \times Z)^{(N-1)}$  paths from the node  $P^1$  to the end  $P^N$ . Our objective is to find the least cost path among them.

node  $i$  with the prior node  $j$  in the path. The definitions of data term and smoothness term are shown in

$$\text{Data}_i(u, v, w) = \begin{cases} 0, & \text{if } i \in \text{CanPs} \\ \text{penalty}D, & \text{if } i \notin \text{NCanPs} \\ \text{penalty}V, & \text{if } i \in \text{VirPs} \end{cases}$$

$$\text{Smoothness}_{i,j}(u, v, w) = \text{penalty}S \times \text{Dis}(i, j) \quad (18)$$

where  $\text{CanPs}$  is the set of voxels filled with candidate points,  $\text{NCanPs}$  is the set of voxels filled with noncandidate points, and  $\text{VirPs}$  is the set of virtual nodes for occlusions.

As shown in Fig. 10, each voxel forms a node to construct LCPM. If the node  $i$  is a voxel filled with candidate points, the data term is 0; if the node  $i$  is a voxel filled with noncandidate points, the data term is  $\text{penalty}D$ ; if there is no voxel between two nodes in the axis direction, we fill this no voxel area with virtual nodes and the data term is  $\text{penalty}V$ .

The smoothness term is calculated by  $\text{penalty}S$  and the Euclidean distance  $\text{Dis}$  between  $i$  and  $j$ . Each refinement path corresponds to a cost  $L_N$  in (17) and the least cost is the optimal solution.

The graph to be optimized is constructed by the candidate points. Our goal is to connect them into the optimal curbs. Assuming that there are  $N$  nodes in the results, as shown in Fig. 11. Node  $P^{i+1}$  is shifted from node  $P^i$  by  $(\Delta x, \Delta y, \Delta z)$ . The shifts are integer vectors and range from 0 to  $X$ , 0 to  $Y$ , and 0 to  $Z$  along the  $x$ -axis,  $y$ -axis, and  $z$ -axis, respectively.

It is infeasible to exhaustively search all  $(X \times Y \times Z)^{(N-1)}$  paths to find the global optimization. The reduction of the

Candidate points in the search space

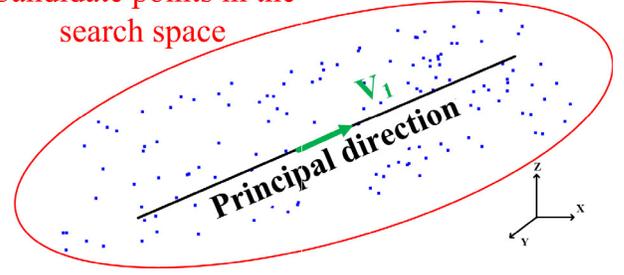
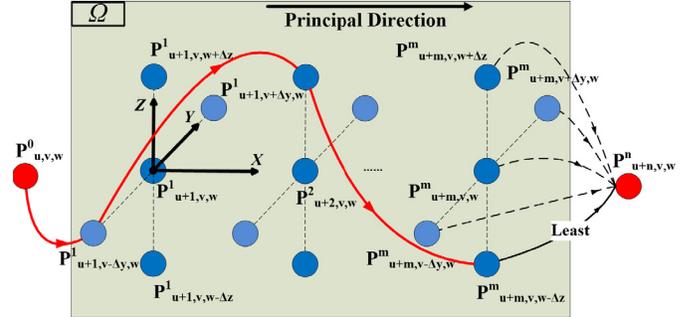


Fig. 12. Principal direction obtained by SVD.


 Fig. 13. New least cost path of graph  $\Omega$  when added a new node  $P^{u+n,v,w}$ .

search space is based on the observation that the least cost path is in the principal direction of the candidate points. Thus, we can search the optimal path along the principal direction progressively.

Since the road is often continuous, the principal direction can be a piecewise constant in the search space. We estimate the principal direction of the path using the singular value decomposition (SVD) method. Assuming that there are  $q$  candidate points in the search area, from SVD, we have

$$\mathbf{D}_{q \times 3} = \mathbf{U}_{q \times q} \mathbf{S}_{q \times 3} \mathbf{V}_{3 \times 3}^T \quad (19)$$

where  $\mathbf{D}$  is the input matrix decomposed into the matrices  $\mathbf{U}$ ,  $\mathbf{S}$ , and  $\mathbf{V}$ . Denote the first, the second, and the third column of  $\mathbf{V}$  as  $\mathbf{V}_1$ ,  $\mathbf{V}_2$ , and  $\mathbf{V}_3$ , respectively. The principal component  $\mathbf{V}_1$ , which corresponds to the largest eigenvalue, is chosen as the principal direction. The obtained principal direction is shown in Fig. 12. The search space is fixed as  $100 \times 100 \times 100$  voxel<sup>3</sup>.

We obtain the step size  $(\Delta x, \Delta y, \Delta z)$  by

$$\left[ \left( 1 - \frac{\|\mathbf{V}_1\|^2}{\sqrt{\|\mathbf{V}_1\|^2 + \|\mathbf{V}_2\|^2 + \|\mathbf{V}_3\|^2}} \right) \cdot (X_c, Y_c, Z_c) \right] \quad (20)$$

where  $X_c$ ,  $Y_c$ , and  $Z_c$  are the length of the  $x$ -axis,  $y$ -axis, and  $z$ -axis in the current search space.

Next, we propose the LCPM to find the optimal path. The path proceeds along the principal direction of the search space, as shown in Fig. 13, where the principal direction is supposed to be the  $x$ -axis.

We add a starting node  $P_{u,v,w}^0$  for each path to be refined. The path is from  $u$  to  $u + m$  in the principal direction. The cost of the connection between the starting node  $P_{u,v,w}^0$  and other nodes is 0. Assuming that we have found the optimal

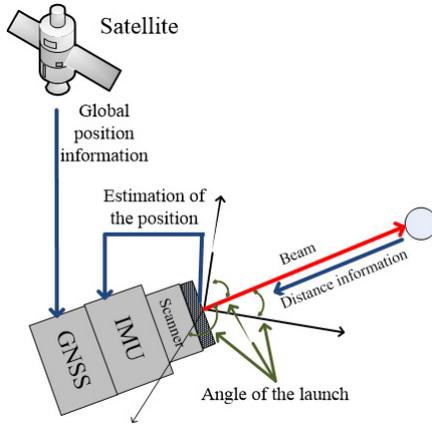


Fig. 14. Process of collecting data.

path from the starting node to each node, now add a new node  $P_{u+n,v,w}^n$  in graph  $\Omega$ . The least cost from  $P_{u+n,v,w}^n$  to its prior node is calculated by (21). For the newly added node  $i$ , the optimization from the starting node to  $i$  contains the solution from the starting node to its prior node  $j$ .  $L_j$  is known, so the computation of  $L_i$  incurs quite a low complexity

$$L_i = \min_{j=1}^{\text{num}} (L_j + \text{Data}_i(u, v, w) + \text{Smoothness}_{i,j}(u, v, w)). \quad (21)$$

We store the least cost path from the starting node to each node. Each path has a cost obtained by (17). We refer to the least cost path to back track the complete path from the end node. Nodes in this least cost path are mapped to voxels and used to refine the incomplete curb edges. Since LCPM takes the connection costs into consideration, it can bring back the noncandidate points to obtain the optimization under the given cost function.

#### IV. EXPERIMENT AND RESULTS

##### A. Data Collection

There are three main components in the mobile LiDAR scanning system, namely, laser scanner, Global Navigation Satellite System (GNSS), and inertial measurement unit (IMU), as shown in Fig. 14. The scanner measures the distance between the system and the object and the angle of the launch is known to calculate the object position. The GNSS locates the global position of the scanner. The IMU is to estimate the position of the scanner when GNSS does not work.

Our mobile LiDAR data are acquired by the Riegl VMX-450 system. This laser scanner uses a narrow infrared laser beam at a very high scanning rate, which can be up to 200 lines/s and enables full 360° beam deflection without any gaps. Our data are collected in a 246142.05 m<sup>2</sup> residential area consisting of various types of roads. The data size is larger than 16.7 GB in “txt” format and contains about 300 million points. The geographic location is from (51°4′15.12” N, 114°5′1.37” W) to (51°4′17.12” N, 114°4′7.47” W). The length is 1166.55 m; the width is 211 m, and the elevation difference is 38.59 m.

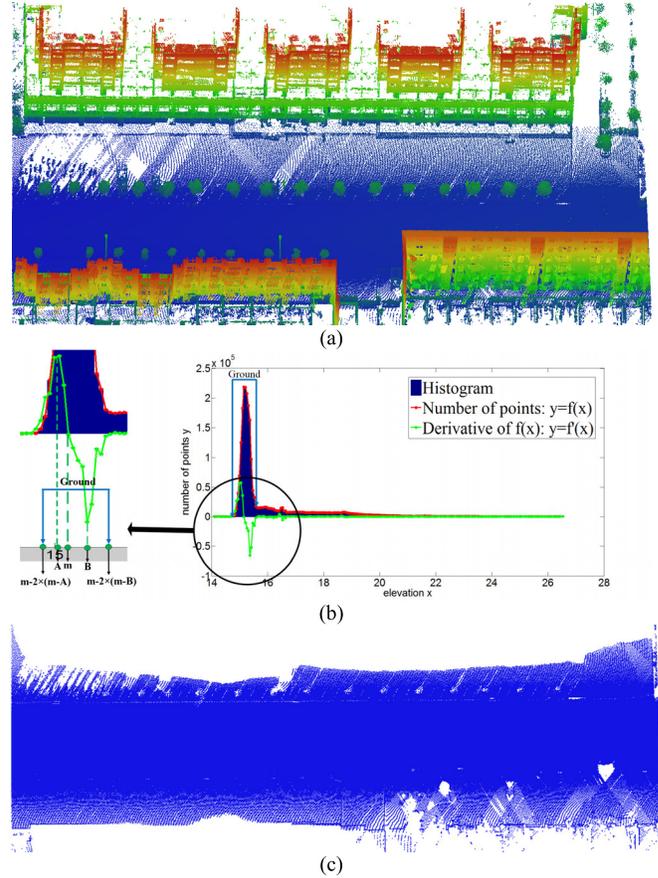


Fig. 15. Process of removing nonground regions. (a) Input point clouds. (b) Elevation histogram. (c) Result of removing nonground areas.

Fig. 15(a) is a piece of the input data. In the preprocessing steps, we remove the nonground areas, such as trees, houses, and cars. As shown in Fig. 15(b), since the road points are much denser than nonground areas, there is a peak in the elevation histogram. The function  $f'(x)$  is the derivative of the function  $f(x)$ , which is used to describe the elevation  $x$  and number of points  $y$ . The global extremal point in  $f(x)$  meets  $f'(x + \varepsilon) \times f'(x - \varepsilon) < 0$ , where  $\varepsilon$  is a small positive number. When  $x$  is equal to  $m$ , the number of points  $y$  achieves the maximum. Two local extremal points in  $f'(x)$  near the point  $x = m$  are  $x = A$  and  $x = B$ . In our algorithm, the elevation from  $m - [2 \times (m - A)]$  to  $m - [2 \times (m - B)]$  is chosen as the ground areas, as shown in Fig. 15(b). The result of removing nonground areas is shown in Fig. 15(c).

We use the voxel-based representation to organize the ground point sets. The volume of each voxel is selected as much as  $0.04 \times 0.04 \times 0.04$  m<sup>3</sup>. The intensity of each voxel is used to calculate the sampling density gradient. Both  $penaltyD$  and  $penaltyS$  depend on the result of the curb point extraction. As shown in Fig. 16, the horizontal axis means the percentage of candidate points in the search space and the vertical axis means the penalty. When the percentage is smaller than 0.04, we think there are no curbs in the current search space. If the extracted candidate points are limited, the penalty  $penaltyD$  should be small in order to consider more noncandidate points whereas the penalty  $penaltyS$  should be

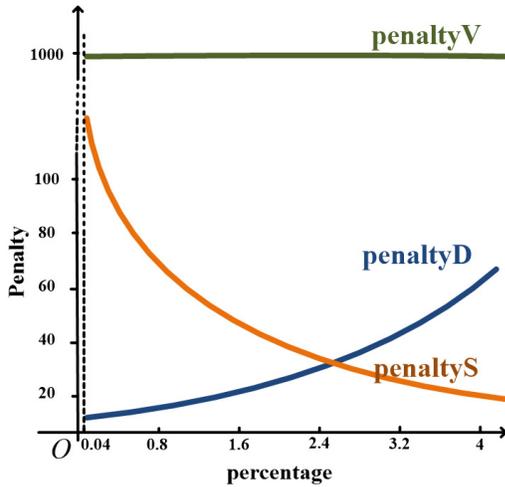


Fig. 16. Selection of the penalty.

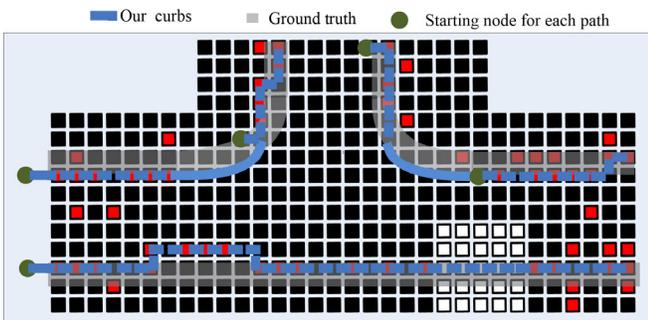


Fig. 17. Process of LCPM.

large to keep the principal direction. If they are sufficient, *penaltyD* should be large enough to consider more candidate points whereas *penaltyS* should be small in case of the zigzag curb. The penalty *penaltyV* should be large, which is 1000 in our algorithm.

The process of LCPM is shown in Fig. 17 and the optimization starts from the left to right. At the intersections, where the principal direction changes greatly, we use a curve for missing curb areas. For occlusions, we find the optimal path based on the virtual nodes.

V. EXPERIMENTS

We evaluate our algorithm in terms of three aspects: robustness, accuracy, and efficiency. To further evaluate the robustness of our algorithm, we test it on two large-scale road environments, including a residential area collected by the Riegl VMX-450 system (16.7 GB, 300 million points) and an urban area collected by the OptechLynx scanner system (1.07 GB, 20 million points).

1) *Extraction of the Curbs*: Fig. 18(a) and (b) shows the projection of curbs on the *XOY* and *xoz* planes, respectively. Most of the existing methods use these projections as the input to extract curbs. As mentioned before, these methods lose all 3-D information and can hardly deal with occlusions. Our algorithm uses the full 3-D information of the point clouds and can deal with the challenging situations caused by either scanner system or complex road environments.

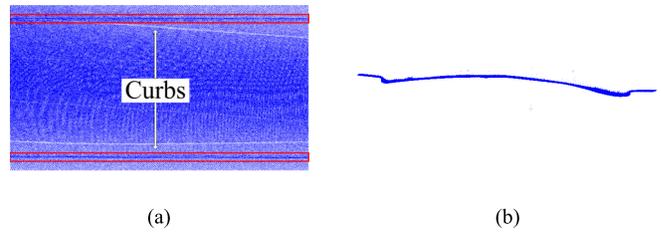


Fig. 18. Descriptions of curbs. (a) Projection on *XOY*. (b) Projection on *XOZ*.

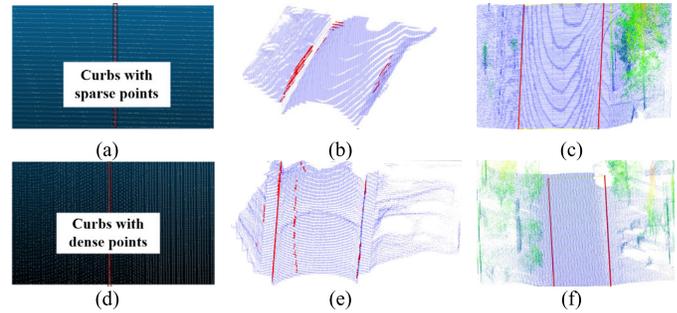


Fig. 19. Uneven curb points. (a) Curbs with sparse points. (b) Extracted candidate points of (a). (c) Refinement of (b). (d) Curbs with dense points. (e) Extracted candidate points of (d). (f) Refinement of (e).

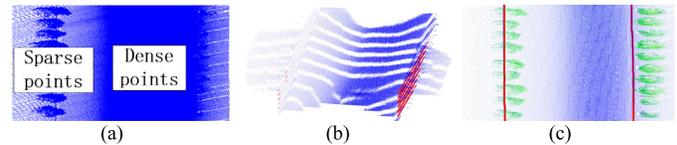


Fig. 20. Various densities. (a) Densities of points are varying from left side to right side of the road. (b) Extracted candidate points of (a). (c) Refinement of (b).

The following are the results of our method on different challenging situations. To better show the results, we highlight results by large red points.

a) *Uneven density*: The curb points may be sparse or dense, as shown in Fig. 19(a) and (d). The extracted candidate points are shown in Fig. 19(b) and (e), respectively. The candidate curbs are noisy and incomplete. By using LCPM, we link them into the optimal curbs, as shown in Fig. 19(c) and (f), respectively.

An example of varying densities between right and left side of the road is shown in Fig. 20. The various densities lead to the unreliable computation of our sampling density gradients. This causes undesirable extraction, as shown in Fig. 20(b). As seen from Fig. 20(c), although the various densities may cause incorrect candidate points extracted, our model still can obtain the desirable curbs.

The density of the point clouds collected from various systems is different. To test our algorithm, we downsample the road point clouds to different cases, as shown in Fig. 21(a)–(d). Results show that our method is robust to the sparsity. Even for the case where the point clouds are downsampled to 1%, the proposed method can still extract the curbs, which is difficult for any existing methods.

One challenging problem, as shown in Fig. 21(a), is that there are gaps in point clouds caused by the MLS itself. These

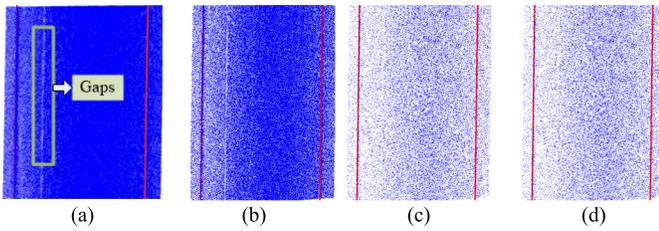


Fig. 21. Sparse points. Sparse road by sampling data in (a) 100%, (b) 50%, (c) 10%, and (d) 1%.

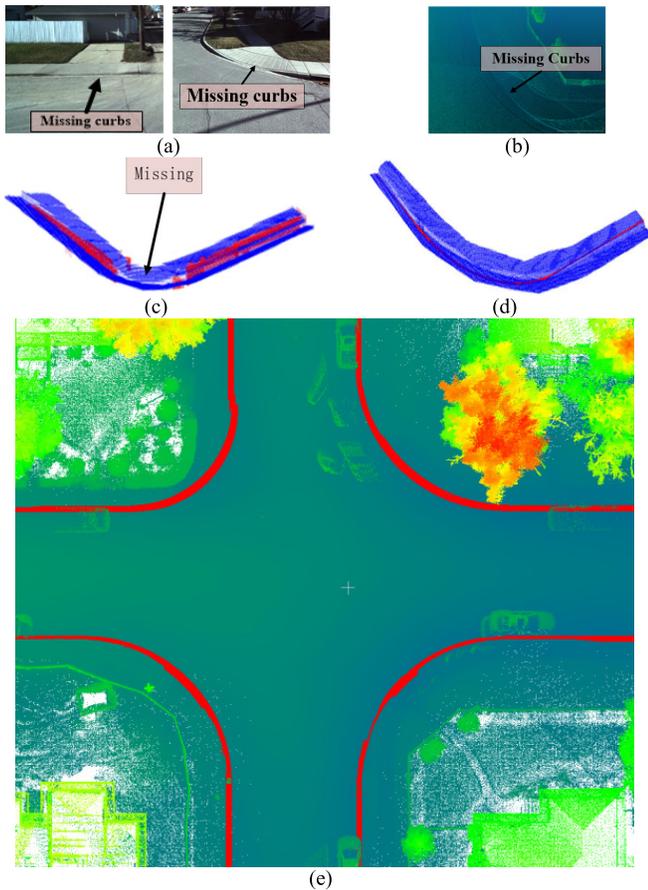


Fig. 22. Missing curbs. (a) Missing curb areas in 2-D image. (b) Missing curb areas in 3-D point clouds. (c) Missing candidate points in curb areas. (d) Our refined result of (c). (e) Results of an entire crossroad.

gaps are easy to be wrongly detected as curbs but well filtered by our method.

*b) Missing curb areas:* The curb on the road may be missing, as shown in Fig. 22(a), which is designed for wheelchairs and bicycles. There is no curb information in these areas, as shown in Fig. 22(b). This is our limitation, because there is only one large sampling density gradient in these areas. The candidate points are missing totally, as shown in Fig. 22(c). If missing curb areas are along the straight road, we can obtain the complete curbs based on the neighbor information. However, if these areas are at the intersection road, we can only use a fixed curve based on the prior knowledge to link curbs, as shown in Fig. 22(d).

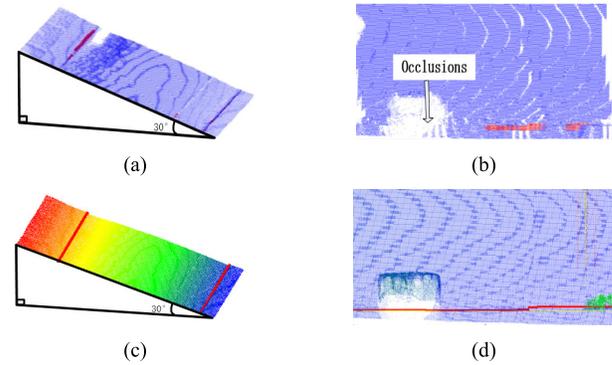


Fig. 23. Slope and occlusion. (a) Sloping road. (b) Occluded road. (c) Results of (a). (d) Results of (b).

Fig. 22(e) is the result of an entity crossroad to demonstrate our results.

*c) Slope and occlusion:* To test the sloping road, we lift one side of the road up to  $30^\circ$ , as shown in Fig. 23(a). This is difficult for extraction algorithms based on the elevation. For occlusions caused by cars or pedestrians on the road, as shown in Fig. 23(b), LCPM considers the virtual nodes to find the optimal path.

The change  $C$ , which is used to calculate the energy, is independent of the coordinate system. Thus, our method is invariant to the rotation, and hence, curbs in the slope are well extracted with our algorithm. Extracted candidate points and refinements are shown in Fig. 23(a) and (c).

As opposed to the missing curb areas, there are no points in occlusions. In our algorithm, occlusions are filled with virtual nodes. To connect the virtual nodes, a large  $penaltyV$  is used. For a small occlusion, the optimal path passes through the missing curb areas, as shown in Fig. 23(d). However, for a large occlusion, we empirically conclude that there is no curb when the percentage of candidate points is lower than 0.04. Extracted candidate points and refinements are shown in Fig. 23(b) and (d).

*d) Large-scale experiments:* The residential area mostly contains trees, parking cars, and houses. Fig. 24(a)–(f) corresponds to six parts of the residential area. As shown in Fig. 24(a), we zoom into three areas to show the results, including the straight curbs (area A), the intersection area (area B), and the occluded area (area C). Curbs in these areas are well extracted and refined.

The urban area mostly contains trees, buildings, and traffic facilities, as shown in Fig. 24(g). We zoom into two areas to show the results, including the occluded area (area A) and an alley (area B). In both the areas, curbs are well extracted.

To visualize the difference between our results and the ground truth, we show that the candidate points, resultant curbs, ground truth, and the overlap in Fig. 25 refer to Fig. 24(a).

Results in Figs. 24 and 25 demonstrate that our method is robust against large-scale data testing. We also use videos to show our results in 3-D in the online supplementary materials.

*2) Quantitative Evaluation:* In this section, we quantify the difference between our results and the ground truth and compare our method with other related work.

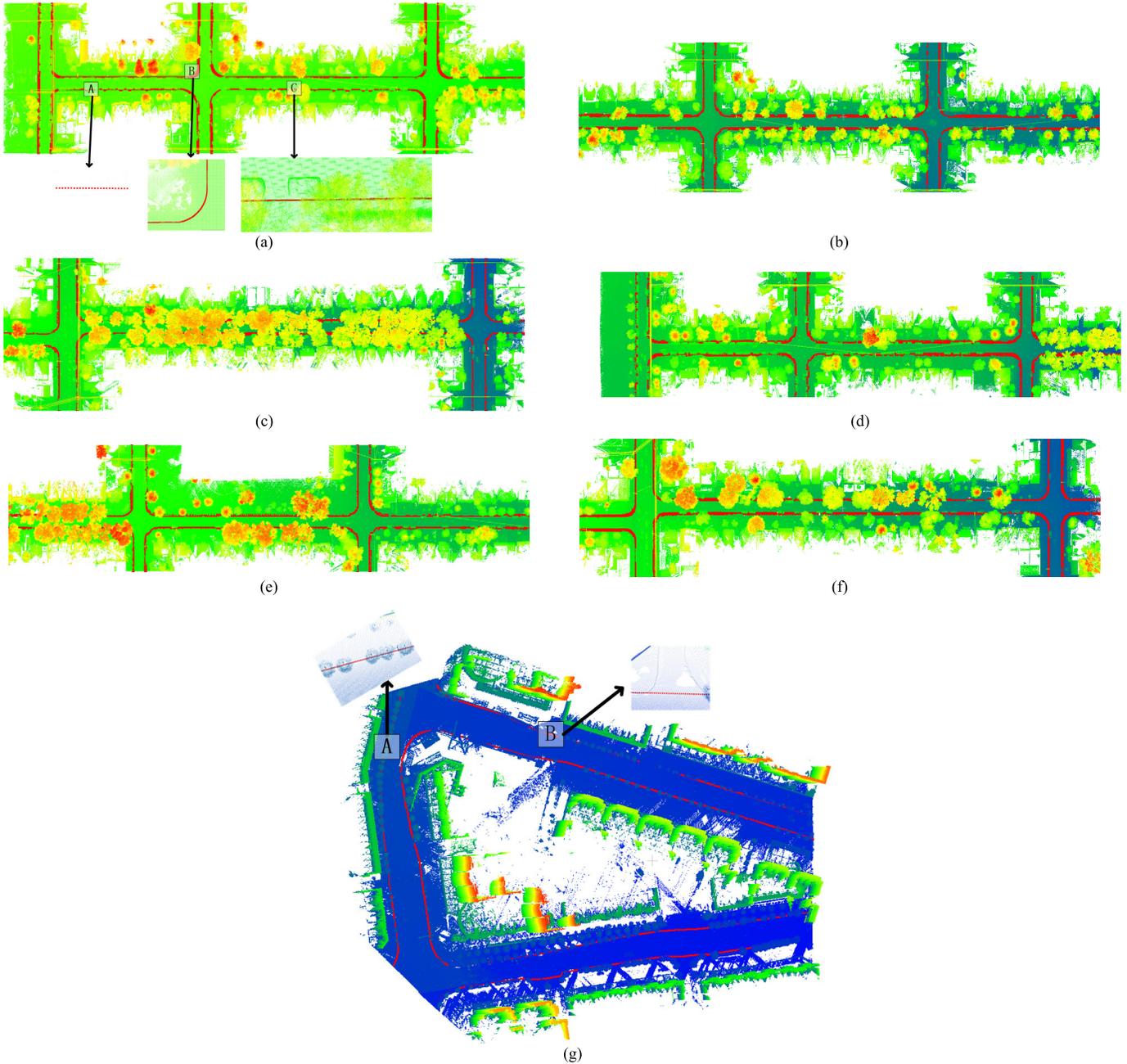


Fig. 24. Experimental results, including, a large residential area and an urban area. (a) Results of Part 1 from the large-scale residential area. (b) Results of Part 2 from the large-scale residential area. (c) Results of Part 3 from the large-scale residential area. (d) Results of Part 4 from the large-scale residential area. (e) Results of Part 5 from the large-scale residential area. (f) Results of Part 6 from the large-scale residential area. (g) Results of the large-scale urban area.

Assuming there is a point  $p$  in the clouds,  $L$  is the curb obtained by our method and  $L'$  is the ground truth obtained by the manual method. For the classification, there are four results of  $p$ , namely, true positive (TP) if  $p \in L \cap L'$ , true negative (TN) if  $p \in \bar{L} \cap \bar{L}'$ , false positive (FP) if  $p \in L \cap \bar{L}'$ , and false negative (FN) if  $p \in \bar{L} \cap L'$ , as shown in Fig. 26. We evaluate the difference in terms of four aspects based on TP, TN, FP, and FN [26], namely, TP rate (TPR), TN rate (TNR), positive predictive value (PPV), and negative and predictive value (NPV).

We need a parameter  $D$  to decide whether the test point  $p$  belongs to  $L$  or  $L'$ . If the distance between  $p$  and  $L$  or  $L'$

is less than  $D$ ,  $p \in L$  or  $p \in L'$ , else  $p \in \bar{L}$  or  $p \in \bar{L}'$ , as shown in Fig. 26. In the following, we show the accuracy under different  $D$  values in Table I.

Table I reports the quantitative results of the above large-scale experiments, where  $SL$ ,  $Int$ , and  $All$ , respectively, denote the straight areas, intersections, and all areas. The extracted curbs consist of 294418 points and 115971 of them belong to straight areas and others are intersections. The ground truth contains 115240 points and 24258 of them belong to straight areas and others are intersections. TPR is  $TP/(TP + FN)$  means the probability of true curbs that can be extracted, which is the completeness of curbs. From Table I, it can be observed that

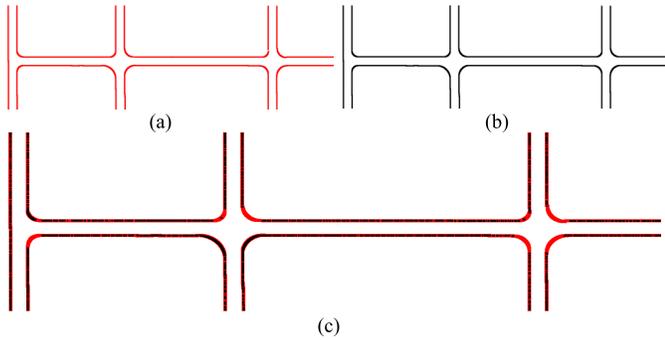


Fig. 25. Visualization of the difference between our results and the ground truth for Part 1. (a) Our results. (b) Ground truth. (c) Overlap between (a) and (b).

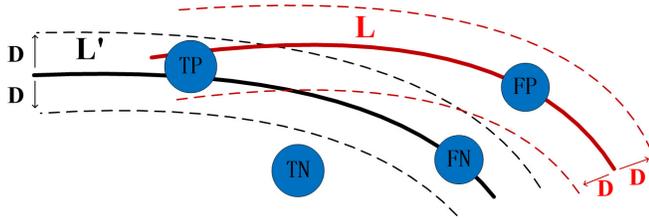


Fig. 26. Four types of results for each point, namely, TP, TN, FP, and FN.  $L$  is the result of our method and  $L'$  is the ground truth. The width of  $L$  or  $L'$  is  $2 \times D$ .

TABLE I  
QUANTITATIVE EVALUATION

Evaluation (%)	D (m)					
	0.4	0.2	0.12	0.08	0.04	
TPR	<i>SL</i>	89.49	81.30	66.55	59.13	26.71
	<i>Int</i>	61.87	41.21	26.06	18.66	8.09
	<i>All</i>	<b>78.62</b>	<b>66.25</b>	<b>51.37</b>	<b>43.63</b>	<b>19.40</b>
TNR	<i>SL</i>	99.77	99.70	99.61	99.56	99.73
	<i>Int</i>	99.52	99.64	99.75	99.82	99.93
	<i>All</i>	<b>99.65</b>	<b>99.67</b>	<b>99.68</b>	<b>99.69</b>	<b>99.83</b>
PPV	<i>SL</i>	91.54	78.93	56.07	36.01	10.22
	<i>Int</i>	69.37	48.84	32.22	21.26	7.46
	<i>All</i>	<b>83.29</b>	<b>69.00</b>	<b>49.15</b>	<b>32.33</b>	<b>9.64</b>
NPV	<i>SL</i>	99.71	99.74	99.75	99.83	99.92
	<i>Int</i>	99.34	99.51	99.67	99.79	99.93
	<i>All</i>	<b>99.52</b>	<b>99.62</b>	<b>99.71</b>	<b>99.81</b>	<b>99.92</b>

when the distance parameter  $D$  is 0.4 m, the completeness of curbs is up to 78.62%. PPV is  $TP/(TP + FP)$  means that the extracted curbs belong to the true curbs, which is the correctness of the curbs. When  $D$  is 0.4 m, the correctness of curbs is up to 83.29%. TNR is  $TN/(FP + TN)$  means the probability of noncurb areas that can be extracted, which is the completeness of noncurbs. NPV is  $TN/(TN + FN)$  means the extracted noncurb points belong to the noncurb areas, which is the correctness of the noncurbs. Table I indicates that at intersections, our method has a poor performance, mainly due to the absence of curb information.

### 3) Comparison With Existing Methods:

a) *Detection methods:* We compare our algorithm with related detection methods, including, Yu *et al.* (EEC: elevation gradient computation, elevation gradient filtering and corner selection) [6], Kellner *et al.* (IEPF: iterative end-point fitting) [10], Rodríguez-Cuenca *et al.* (TML: thresholding, morphological

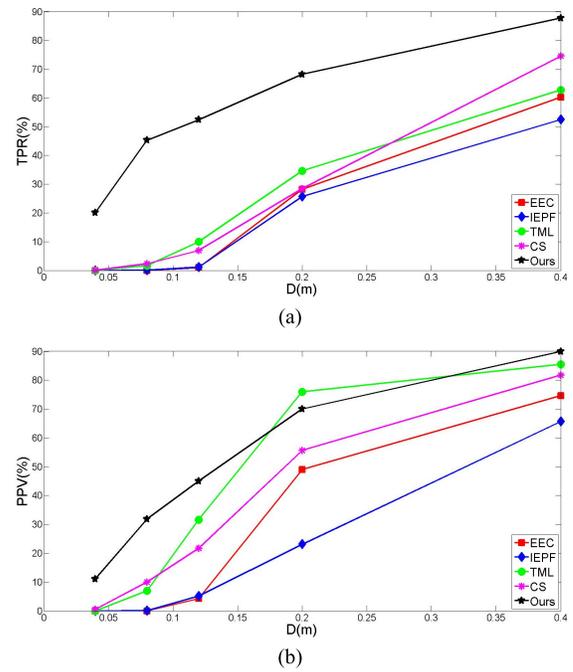


Fig. 27. Comparison with the existing curb detection methods, including, EEC, IEPF, TML, and CS. (a) TPR to evaluate the completeness. (b) PPV to evaluate the correctness.

processing, and linear feature detection) [16], and Yang *et al.* (CS: curb structure) [11], using the TPR and PPV to evaluate the completeness and correctness of curbs.

There are three steps in EEC, namely, elevation gradient computation, elevation gradient filtering, and curb corner point selection. This method is based on the elevation filtering, which fails when curbs are in quite different elevations or occluded. The IEPF uses iterative end-point fitting algorithm to segment the scenes. This algorithm relies on the elevation of curb and the flatness of the road to detect the curbs, which is difficult to work in complex urban scenes. TML uses the projection to detect curbs by three steps, namely, thresholding, morphological processing, and linear feature detection. Each step needs many parameters, which are difficult to choose. CS is proposed to detect road curbs by a predefined curb model, which is based on the elevation jump, point density, and slope change.

We compare with the above-mentioned methods using the data of Part 1. As shown in Fig. 27, our method is much better than the above-mentioned methods in detecting curbs, especially when  $D$  is small. The accuracy of TML can be high, but it relies on the parameters heavily, which is difficult to tune.

b) *Refinement methods:* We also compare our method with some typical refinement methods, including, LS, HT, and RANSAC. We test the robustness of our method against noise by adding a random number from  $-T \times d$  to  $T \times d$  to the coordinate of each point, where  $d$  is the minimum point distance between two points and  $T$  is to set the range.

For a small random noise ( $T = 2$ ), the candidate points extracted by our method are shown in Fig. 28(a). Increasing the level of the random noise ( $T = 4$ ), we achieve an

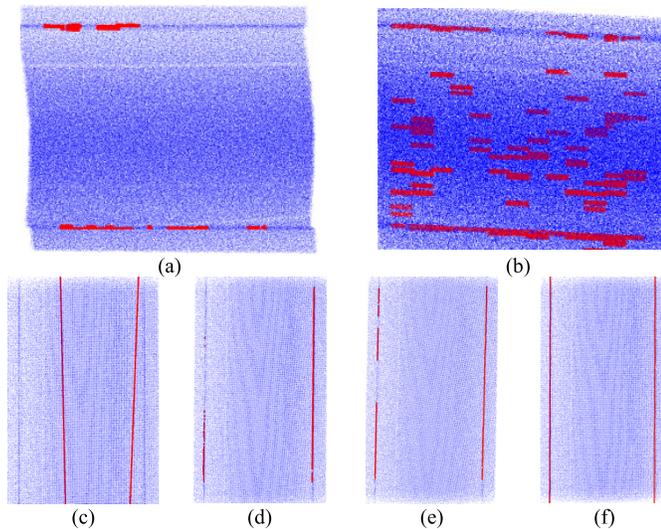


Fig. 28. Comparison with the existing refinement methods. (a) Extracted candidate points from a small random noise ( $T = 2$ ). (b) Extracted candidate points from a large random noise ( $T = 4$ ). (c) Refinement by LS. (d) Refinement by HT. (e) Refinement by RANSAC. (f) Refinement by LCPM.

undesirable result, as shown in Fig. 28(b). These candidate points, containing outliers, are the input for the later refinement.

The results of LS, HT, and RANSAC and our method are shown from Fig. 28(c)–(e), respectively. LS fails to refine curbs due to incorrect mean point by outliers, and HT and RANSAC obtain acceptable curbs from our candidate points. 3-D HT obtains the curb correctly while requires the best line in the exactly the same plane, which is not always the case in point clouds. For RANSAC, only if all inliers are obtained, the line can be fitted accurately. Moreover, all these typical refinement methods do not consider the cost to connect each point. If there are few candidate points or few straight lines, both HT and RANSAC can hardly extract curbs. In this condition, LCPM obtains the optimal curbs by considering both candidate and noncandidate points, which enhances the robustness considerably, as shown in Fig. 28(f).

4) *Computational Complexity*: Normally, there are three steps in each algorithm, namely, the generation of the region of interest (ROI), the extraction of candidate curb points, and the refinement of incomplete curbs. The complexity of the extraction in each algorithm is described in the following.

EEC relies on the curb profiles vertical to the road surface and 10–20 cm above the road to generate ROI. The generation uses a thresholding method ( $O(N)$ ). This is followed by the extraction of the candidate points, including, elevation gradient computation ( $O(N)$ ), elevation gradient filtering ( $O(N)$ ), and curb points selection ( $O(N)$ ). Finally, the refinement as mentioned in their paper can be LS ( $O(N^3)$ ).

IEPF calculates the distance between each point and a straight line, obtained by connecting the start and end point of a scan line, to generate the lateral distance ( $O(N)$ ) in the preprocessing. The points are segmented by the lateral distance and classified by the clustering algorithms [27] ( $O(N^2)$ ). Then, the authors use the decision tree [ $O(R \times N)$ , where

TABLE II  
COMPLEXITY OF EACH ALGORITHM

Method	Preprocessing	Extraction	Refinement
EEC	$O(N)$	$O(N)$	$O(N^3)$
IEPF	$O(N)$	$O(N^2)$	$O(N)$
TML	$O(N)$	$O(N^2)$	$O(N)$
CS	$O(N)$	$O(N^2)$	$O(N^2)$
Ours	$O(N)$	$O(N)$	$O(N^3)$



Fig. 29. Overlay of our results on the Google Earth.

$R$  is the depth of the tree] based on three properties, namely, mean height ( $O(N)$ ), angle ( $O(N)$ ), and variance ( $O(N)$ ), to obtain the curbs. The refinement is based on the trajectory of the car. The similar candidate curbs, which show similarity in the lateral distance, are connected in linear time ( $O(N)$ ).

TML uses elevation thresholding to obtain the ROI ( $O(N)$ ). The maximum and minimum thresholds are set to avoid the extraction of points from the road. Then, the morphological processing methods are used to obtain the candidate curbs [no less than  $O(2 \times N^2)$ ], including, erosion to remove the isolated points and dilation to increase the curb candidate set of points. Refinement is based on the line feature, which is calculated by thresholding the percentage of rows or columns higher than a preset percentage, to connect the candidate points ( $O(N)$ ). Then, the authors perform a rotation around the axis to determine the edges based on the trajectory of the vehicle ( $O(N)$ ).

CS groups point clouds into road cross sections ( $O(N)$ ) based on GPS time in the preprocessing step. Then, they use a sliding window, to extract candidate road areas based on the fact that the road points at one cross section have the identical elevation ( $O(N)$ ), to obtain ROI. The authors detect curbs based on their three proposed models, including, elevation jump ( $O(N)$ ), point density ( $O(N^2)$ ), and slope change ( $O(N)$ ). The refinement of the candidate curbs includes using  $K$ -nearest neighbor to cluster them ( $O(N^2)$ ), removing fake curbs that contain few points ( $O(N)$ ), and connecting the curbs that are sorted along the direction of the curbs ( $O(N)$ ).

Our method calculates the histogram of the elevation to generate ROI ( $O(N)$ ) and then calculate sampling density gradients in each axis direction to obtain the energy for each point ( $O(3 \times N)$ ) followed by the refinement LCPM ( $O(N^3)$ ).

We show all the computational complexities in Table II. From Table II, the complexity of ours is the same as existing methods in the preprocessing step and much lower than IEPF, TML, and CS in the extraction. EEC has a low complexity, because it only depends on the unreliable elevation difference. For the refinement, our complexity is higher, because we do not use any extra information, such as trajectory for IEPF and TML, or GPS time for CS. However, only our refinement method achieves global optimization.

TABLE III  
LENGTH AND WIDTH OF THE SELECTED SIX AREAS

Map (m)		Ours (m)		Geographic location
L	W	L	W	
108	87	110.93	86.74	A <sub>0</sub> (51.071492N,-114.08303W), A <sub>1</sub> (51.072267N,-114.08152W)
157	86	158.57	85.58	B <sub>0</sub> (51.071506N,-114.08128W), B <sub>1</sub> (51.072267N,-114.07904W)
158	88	159.94	85.67	C <sub>0</sub> (51.071516N,-114.07884W), C <sub>1</sub> (51.072276N,-114.07657W)
157	85	158.69	86.10	D <sub>0</sub> (51.071498N,-114.07637W), D <sub>1</sub> (51.072269N,-114.07414W)
157	84	157.33	83.87	E <sub>0</sub> (51.071510N,-114.07392W), E <sub>1</sub> (51.072262N,-114.07167W)
268	87	270.65	87.61	F <sub>0</sub> (51.071584N,-114.07143W), F <sub>1</sub> (51.072362N,-114.06756W)

5) *Results on the 2-D Map:* We overlay our results on the images from Google Earth. As shown in Fig. 29, our results can match the curbs in the map accurately.

To quantify our results on 2-D map, we compare the length ( $L$ ) and width ( $W$ ) of selected six places in Google Earth with our results in Table III.

From Table III, the difference between our results from the above detected curbs and the Google Earth is 0.006/m and the mean square error is 2.67. These evaluations show that our results are accurate and reliable.

## VI. CONCLUSION

Curb extraction is essential for understanding road environments. This paper presents a robust, accurate, and efficient solution for road curb extraction from mobile LiDAR point clouds. To the best of our knowledge, this is the most comprehensive work on road curb extraction from point clouds. We evaluate the proposed method on a large-scale residential area and an urban area. Our algorithm works effectively for large-scale mobile LiDAR point clouds. Different quantitative evaluations, including, the TPR, TNR, PPV, and NPV, indicate that our method is more accurate than the existing algorithms.

Possible directions for the future research include segmentation of scenes, classification of objects, and understanding of traffic environments.

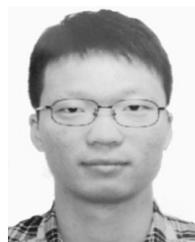
## ACKNOWLEDGMENT

The authors would like to thank the City of Calgary for providing the mobile LiDAR data. They would also like to thank Dr. J. Peethambaran for proofreading this paper.

## REFERENCES

- [1] R. Wang, "Towards urban 3D modeling using mobile LiDAR and images," M.S. thesis, Dept. Elect. Comput. Eng., McGill Univ., Montreal, QC, Canada, 2011.
- [2] G. Vosselman and L. Zhou, "Detection of curbstones in airborne laser scanning data," *Int. Arch. Photogram., Remote Sens. Spatial Inf. Sci.*, vol. 38, pp. 111–116, Sep. 2009.
- [3] L. Zhou and G. Vosselman, "Mapping curbstones in airborne and mobile laser scanning data," *Int. J. Appl. Earth Observat. Geoinf.*, vol. 18, pp. 293–304, Aug. 2012.
- [4] G. Zhao and J. Yuan, "Curb detection and tracking using 3D-LIDAR scanner," in *Proc. 19th IEEE Int. Conf. Image Process.*, Sep. 2012, pp. 437–440.

- [5] A. Y. Hata, F. S. Osorio, and D. F. Wolf, "Robust curb detection and vehicle localization in urban environments," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2014, pp. 1257–1262.
- [6] 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.
- [7] Y. Chen *et al.*, "Using prior shapes in geometric active contours in a variational framework," *Int. J. Comput. Vis.*, vol. 50, no. 3, pp. 315–328, Dec. 2002.
- [8] F. Oniga and S. Nedeveschi, "Polynomial curb detection based on dense stereovision for driving assistance," in *Proc. 13th Int. IEEE Conf. Intell. Trans. Syst. (ITSC)*, Sep. 2010, pp. 1110–1115.
- [9] Y. He, "Automated 3D building modelling from airborne LiDAR data," Ph.D. dissertation, Dept. Infrastruct. Eng., Univ. Melbourne, Melbourne, VIC, Australia, 2015.
- [10] M. Kellner, U. Hofmann, M. E. Bouzouraa, H. Kasper, and S. Neumaier, "Laserscanner based road curb feature detection and efficient mapping using local curb descriptions," in *Proc. IEEE 17th Conf. Intell. Trans. Syst.*, Oct. 2014, pp. 2602–2609.
- [11] 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. Photogram. Remote Sens.*, vol. 79, pp. 80–93, May 2013.
- [12] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vis.*, vol. 1, no. 4, pp. 321–331, 1988.
- [13] H. Mayer, I. Laptev, and A. Baumgartner, "Multi-scale and snakes for automatic road extraction," in *Proc. ECCV*, 1998, pp. 720–733.
- [14] R. Marikhu, M. N. Dailey, S. Makhanov, and K. Honda, "A family of quadratic snakes for road extraction," in *Proc. ACCV*, 2007, pp. 85–94.
- [15] T. Peng, I. H. Jermyn, V. Prinet, and J. Zerubia, "An extended phase field higher-order active contour model for networks and its application to road network extraction from VHR satellite images," in *Proc. ECCV*, 2008, pp. 509–520.
- [16] B. Rodríguez-Cuenca, S. García-Cortés, C. Ordóñez, and M. C. Alonso, "An approach to detect and delineate street curbs from MLS 3D point cloud data," *Autom. Construct.*, vol. 51, pp. 103–112, Mar. 2015.
- [17] P. Kumar, C. P. McElhinney, P. Lewis, and T. McCarthy, "An automated algorithm for extracting road edges from terrestrial mobile LiDAR data," *ISPRS J. Photogram. Remote Sens.*, vol. 85, pp. 44–55, Nov. 2013.
- [18] A. Boyko and T. Funkhouser, "Extracting roads from dense point clouds in large scale urban environment," *ISPRS J. Photogram. Remote Sens.*, vol. 66, no. 6, pp. S2–S12, 2011.
- [19] P. Zhu, Z. Lu, X. Chen, and A. Eiumnroh, "Extraction of city roads through shadow path reconstruction using laser data," *Photogram. Eng. Remote Sens.*, vol. 70, no. 12, pp. 1433–1440, Dec. 2004.
- [20] X. Hu, C. V. Tao, and Y. Hu, "Automatic road extraction from dense urban area by integrated processing of high resolution imagery and LIDAR data," in *Proc. Int. Arch. Photogram., Remote Sens. Spatial Inf. Sci.*, vol. 35, Istanbul, Turkey, Jul. 2004, p. B3.
- [21] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. 4th Alvey Vis. Conf.*, 1988, pp. 147–151.
- [22] H. Farid and E. P. Simoncelli, "Optimally rotation-equivariant directional derivative kernels," in *Proc. Comput. Anal. Images Patterns*, Aug. 2005, pp. 207–214.
- [23] G. H. Golub and C. F. van Loan, "An analysis of the total least squares problem," *SIAM J. Numer. Anal.*, vol. 17, no. 6, pp. 883–893, 1980.
- [24] J. Knopp, M. Prasad, G. Willems, R. Timofte, and L. van Gool, "Hough transform and 3D SURF for robust three dimensional classification," in *Proc. ECCV*, 2010, pp. 589–602.
- [25] R. Schnabel, R. Wahl, and R. Klein, "Efficient RANSAC for point-cloud shape detection," *Comput. Graph. Forum*, vol. 26, no. 2, pp. 214–226, Jun. 2007.
- [26] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006.
- [27] R. O. Duda, *Pattern Classification and Scene Analysis*, 1st ed. New York, NY, USA: Wiley, 1973.



**Sheng Xu** received the B.S. and M.S. degrees in computer science and technology from Nanjing Forestry University, Nanjing, China, in 2010 and 2012, respectively. He is currently pursuing the Ph.D. degree with the Department of Geomatics Engineering, University of Calgary, Calgary, AB, Canada.

His research interests include computer vision, mobile LiDAR data processing, and scene understanding from 3-D point clouds.



**Ruisheng Wang** received the B.Eng. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, the M.Sc.E. degree in geomatics engineering from the University of New Brunswick, Fredericton, NB, Canada, and the Ph.D. degree in electrical and computer engineering from McGill University, Montréal, QC, Canada.

He has been an Industrial Researcher with Nokia, Chicago, IL, USA since 2008. He joined the Department of Geomatics Engineering, University of Calgary, Calgary, AB, Canada in 2012 as an

Assistant Professor. His current research interests include geomatics and computer vision.



**Han Zheng** received the B.S. and M.S. degrees in geodesy and geomatics engineering from Wuhan University, Wuhan, China, in 2011 and 2013, respectively. He is currently pursuing the master's degree in digital image systems with the Department of Geomatics Engineering, University of Calgary, Calgary, AB, Canada.

His research interests include laser measurement application, computer vision, digital image processing, and object recognition from 3-D point clouds.