ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/277952414

Velodyne-based Curb Detection Up to 50 Meters Away

Research · June 2015

DOI: 10.13140/RG.2.1.4714.7680

CITATION
1

READS
668

1 author:

Tongtong Chen
National University of Defense Technology

27 PUBLICATIONS **158** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

I am a engineer in Beijing View project

Tracking View project

All content following this page was uploaded by Tongtong Chen on 09 June 2015.

The user has requested enhancement of the downloaded file.

Velodyne-based Curb Detection Up to 50 Meters Away

Tongtong Chen[1] and Bin Dai[1] and Daxue Liu[1] and Jinze Song[2] and Zhao Liu[1]

***Abstract*— Long range curb detection is crucial for an Autonomous Land Vehicle (ALV) navigation in urban environments. This paper presents a novel curb detection algorithm which can detect the curbs up to 50 meters away with Velodyne LIDAR. Instead of building a Digital Elevation Map (DEM) and utilizing geometric features (like normal direction) to extract candidate curb points, we take each scan line of Velodyne LIDAR as a processing unite directly. Some feature points, which are extracted from individual scan lines, are selected as the initial curb points by the distance criterion and Hough Transform (HT). Eventually, iterative Gaussian Process Regression (GPR), which utilizes the above initial curb points as the initial seeds, is exploited to represent both the curved and straight-line curb model. In order to verify the effectiveness of our algorithm quantitatively, 2934 Velodyne scans are collected in various urban scenes with our ALV, and 566 of them are labelled manually[1]. Our algorithm is also compared with two other curb detection techniques. The experimental results on the dataset show promising performance.**

I. INTRODUCTION

Navigation in urban environments is a challenging task for ALVs. However, it also has broad application prospects both in national defence and civilian traffic fields. Curbs, which delimit the boundaries of the roads, can provide rich information about the perception and local path planning for ALVs. For instance, the moving vehicles only appear on the road between the left and right curbs while most of the pedestrians are on the sidewalk; the curbs can also provide the types and accurate locations of the road intersections[1], [2], which is very essential to local path planning. Moreover, because of the occlusion of the trees and high-rises in urban environments, the curb has become a commonly-used feature for position estimation recently[3], [4]. Fig. 1 shows a typical urban scene with curve curbs.

This paper presents a novel curb detection algorithm for an ALV equipped with Velodyne HDL-64E LIDAR. We begin with the extraction of feature points from individual scan lines; then the distance criterion and HT are adopted to select the initial curb points from these feature points; eventually, iterative GPR, which utilizes the initial curb points as the initial seeds, is used to represent the curb models. Because of the powerful basis of GPR for modelling spatially correlated data, our approach can handle both the straight-line and curved curbs. Additionally, instead of building a DEM with all 3D points in one scan (as [5]), we directly extract feature points from individual scan lines, and then use the regression method to represent the curb models. Both the local information in individual scan lines and the global information among all the scan lines are fully utilized, which makes our approach detect the curbs up to 50 meters away.

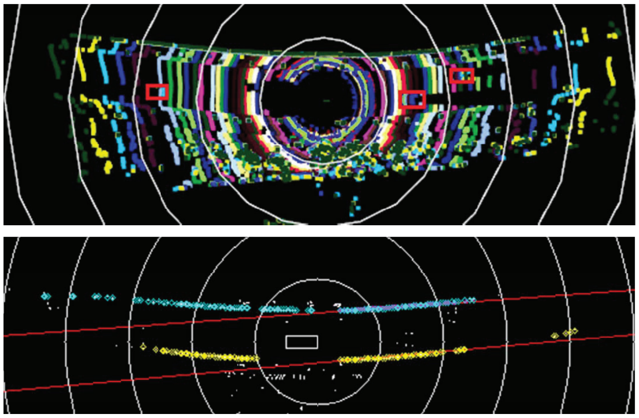

Fig. 1. A typical urban environment with curved road curbs and three dynamic vehicles (red rectangles) in the top image. Each color represents a scan line of the Velodyne HDL-64E LIDAR. The image below shows the results of our algorithm. The yellow and cyan points are right and left curb points respectively. Each white circle represents 10 meters away.

The main contribution of this paper are two-folds. The most significant one is that feature points are extracted from individual scan lines, and iterative GPR, which has both powerful approximate and outlier rejection abilities, is utilized to represent both straight-line and curved curb models. It is crucial to ALVs running on curved roads in urban environments. The second contribution is that 566 Velodyne scans collected in various urban scenes are labelled manually for curb detection. Still to our best knowledge, this is the first public dataset for curb detection with Velodyne HDL-64E LIDAR.

The rest of the paper is structured as follows. In the next section, an outline of the related work is described. Section 3 presents the steps of the curb detection algorithm in detail. Some experimental results are shown in Section 4. Our algorithm is compared with two other curb detection techniques on the public dataset. Finally, conclusion is given in Section 5.

II. RELATED WORK

With the cameras and LIDAR sensors becoming standard equipment in ALVs, much research has been carried out

*This work is supported by National Nature Science Foundations under Grant 61075043 and 61375050

[1]Tongtong Chen, Bin Dai, Daxue Liu and Zhao Liu are with College of Mechatronic Engineering and Automation, National University of Defense Technology, Changsha 410073, Hunan, China `bindai.cs@gmail.com`

[2]Jinze Song is with the Graduate School, National University of Defense Technology, Changsha 410073, Hunan, China `nwsac97@gmail.com`

[1] All the 566 labelled scans with INS data can be download from http://dl.vmall.com/c0qc2yjhkn

during the past decades concerning curb detection. Generally, these methods can be divided into three groups according to their sensors: the methods based on stereo vision[6], [7], [8], [9], [10]; LIDAR [3], [5], [11], [18], [12], [13], [14], [15], [16], [17] and camera/LIDAR combination [19], [20].

Siegemund et al. [6] exploited the dense stereo data to detect the straight-line and curved curbs. A temporally integrated conditional random filed is utilized to classify the 3D points into roads or sidewalks; then the parameters of the curbs are estimated by these assigned 3D points. Oniga et al. [8], [7] built a rectangle DEM with the dense 3D stereo data; then HT and relevant cubic polynomials are adopted to extract straight-line and curved curbs from persistent edge points respectively. Padeep et al. [9] proposed Tensor Voting to estimate the normal direction of each pixel, which is used to partition the pixel into curb point or ground point. Kellne et al. [10] introduced a curb detection method based on a DEM with a dense stereo camera. The road curbs are detected based on the calculation of the most probable path, and represented by polygonal chains which can be obtained by the minimal distance approach.

Almost all the curb detection algorithms based on stereo vision need to compute image disparity to get the depth information, and then project these 3D points into a DEM. Thus, they can only detect nearby curbs and are highly sensitive to the illumination and shadows.

Compared to cameras, LIDAR sensors can provide range information directly. Hata et al. [3] utilized ring compression analysis technique to obtain curb candidates, then differential filter, distance filter and regression filter are used to remove false positives in a circular grid. Zhao et al. [5] presented a 3D LIDAR-based curb detection algorithm. The candidate curb points are extracted with three spatial criterions, then parabola model and particle filter are exploited to fit and track the curbs respectively. Stuckler et al. [11] used Velodyne LIDAR as the main sensor for curb detection. They transformed the measurements into a dense image-like representation and used a simple oriented edge filter to detect curbs. Peterson et al. [18] proposed wavelet-based detector for single scan line of Multiple LIDAR to detect the curbs.

Liu et al. [12] proposed a curb detection algorithm with 2D laser range finder. The slope, height difference and height variance features are adopt to extract candidate curb points in a DEM, and 1D GPR is exploited to represent both the straight-line and curved curbs. Hervieu et al. [13] introduced a novel prediction/estimation process for the recognition of the curbs and curb ramps with two RIEGL. The angle between the normal directions of a point and the ground is considered as a significant feature to classify the point into candidate curb or not. Zhang et al. [14] proposed a road and road curb detection algorithm based on 2D laser. The Gaussian differential filter is adopted to identify the curb candidates. Wijesoma et al. [15] exploited extended Kalman Filter to detect and track the road curb from 2D laser scan. Kodagoda et al. [16] utilized 2D laser to detect curbs, which are then tracked by a novel interacting-multiply-model-probabilistic-data-association-filter (IMMPDAF). The IMMPDAF can handle the disappearing, reappearing and maneuvering in clutter environments during the tracking procedure. Kellner et al. [17] introduced a new grid map with local linear descriptors of the curbs in each cell, and the parameters of the linear model are directly adopted to chain the curb features to road curbs.

As the measurements from Velodyne LIDAR quickly become sparse at longer ranges, the curb detection algorithms, which exploit the normal directions or height differences in cells as the features to extract candidate points in the DEM, cannot get long range detection results. As the 2D LIDAR can only obtain one scan line per time, these curb detection algorithms are so sensitive that they can be easily affected by observation noises and complex structures of roads.

Additionally, Kodagoda et al. [19] introduced a promising curb tracking and estimation approach with camera and 2D laser data. Tan et al. [20] proposed a curb detection algorithm based on the curb's geometric property with a color camera and Velodyne LIDAR, which can detect the curbs up to 30 meters away.

Our algorithm is inspired by [18], [21]. We directly extract candidate curb points according to some local features from individual scan lines of Velodyne LIDAR. It can make our algorithm detect the curbs up to 50 meters away. Based on our previous work [12], iterative GPR is also utilized to represent both the straight-line and curved curb models with Velodyne LIDAR.

III. Details of Curb Detection Algorithm

In ALVs, Velodyne HDL-64E LIDAR is becoming a popular sensor with 64 lasers, because it can provides accurate and high frequency 3D measurements. In this paper, the points that are collected within one scan (required 0.1 s) are treated as if they are collected at one time point. Each scan S_t collected at time point t contains 64 scan lines $P_0, \cdots, P_{63}$. In this section, We will take $P_m, m = 0, \cdots, 63$ as processing unit respectively and introduce the curb detection algorithm in detail. The framework is illustrated in Fig. 2.

A. Interpolation of Scan Line

As the curb surface is vertical to the ground and parallels with the driving direction, we would obtain no valid measurements on curb surface at long range at times if it reflects the laser light. In order to obtain good features for every point in the scan line, especially the points on the curbs, it will be beneficial if all neighbours of that point are valid. For the scan line $P_m = \{p_{m,0}, \cdots, p_{m,n}\}$, which contains some invalid measurements, the interpolation fills them in a linear manner (1). The maximum size of the consecutive invalid measurements is limited to νP and the maximum interpolation step per point is νi meters.

$$p_{m,j} = \begin{cases} p_{m,i} + step \cdot (j - i) & \text{if } k - i < \nu P \cap step < \nu i \\ \mathbf{0} & \text{otherwise} \end{cases}, \tag{1}$$

where $step = \frac{\|p_{m,k} - p_{m,i}\|}{k-i}$, $i < j < k$, and $p_{m,j}$ is an invalid measurement while $p_{m,i}$ and $p_{m,k}$ are valid ones.

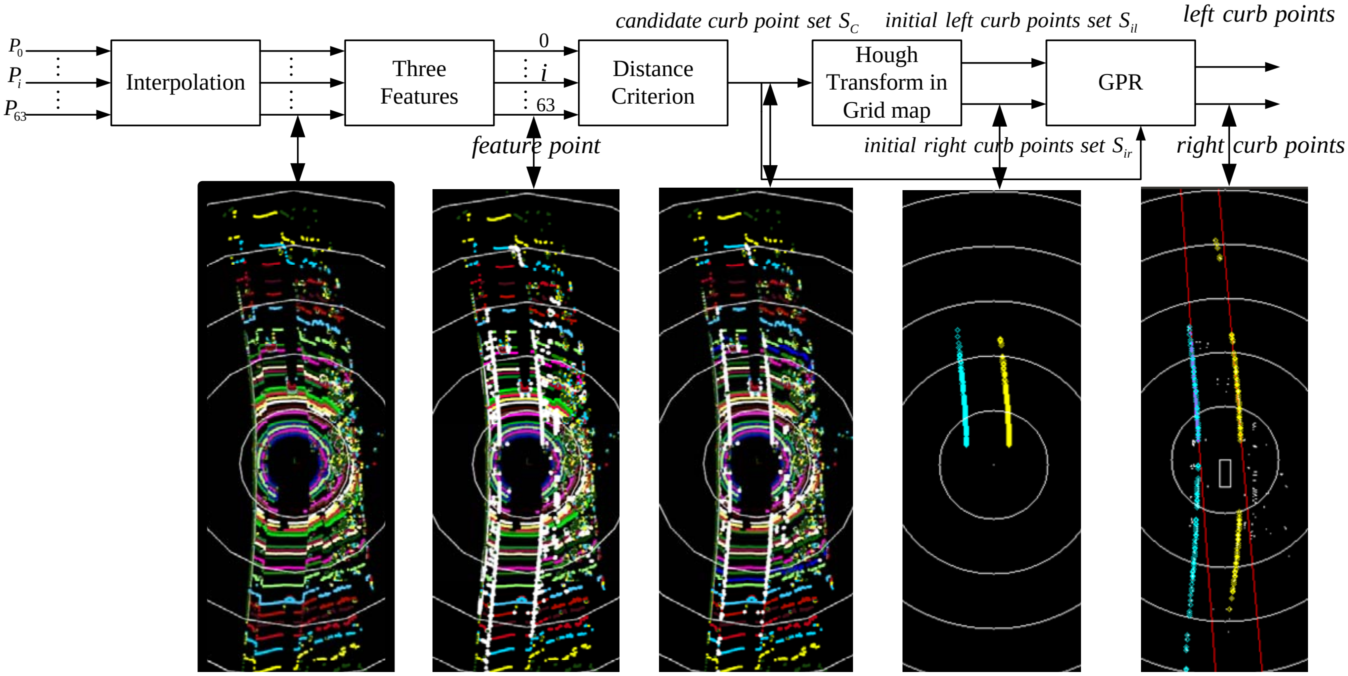
P_0
P_i
P_{63}
Interpolation
Three Features
0
i
63
feature point
Distance Criterion
candidate curb point set S_C
Hough Transform in Grid map
initial left curb points set S_{il}
initial right curb points set S_{ir}
GPR
left curb points
right curb points

Fig. 2. The framework of the proposed curb detection algorithm. The images below are the results. In the second and third images, the white points are the feature points and candidate curb points, respectively. The cyan and yellow points in the fourth and fivth images represent the left and right curbs.

B. Features of Points in Every Scan Line

For every point in scan line P_m, three local features below will be extracted. According to these features, we will then obtain some feature points from individual scan lines.

1) Smoothness Feature: The smoothness feature is a term to describe the smoothness of the area around some point. It was proposed by Zhang et al. [22]. For point $p_{m,i} \in P_m$, let S be the set of consecutive points of $p_{m,i}$. Half of the points in S are on the left side of $p_{m,i}$, and the others are on the right side. The smoothness feature is defined as (2):

$$s = \frac{1}{|S| \cdot \|p_{m,i}\|} \cdot \| \sum_{\substack{p_{m,j} \in S \\ j \neq i}} (p_{m,i} - p_{m,j}) \|, \qquad (2)$$

where $|S|$ is the cardinality of set S. Point $p_{m,i}$ can be classified as a candidate feature point only its s value is larger than a threshold T_0, otherwise, it is a smooth point.

2) Smooth Arc Length Feature: We assume that both the road and sidewalk are the smooth surfaces in urban environments. Thus, at least one side of the candidate curb point should have some consecutive smooth points. Thus, the smooth arc length feature is proposed in this paper to represent the length of the arc which is comprised of the smooth points on each side of the point. Let $S_{smooth-l} = \{p_{m,j}, \cdots, p_{m,i-1}\}$ and $S_{smooth-r} = \{p_{m,i+1}, \cdots, p_{m,k}\}$ be the left and right sets of consecutive smooth points of $p_{m,i}$, respectively. the smooth arc length feature of point $p_{m,i}$ is defined as (3):

$$Al = \begin{bmatrix} \alpha \frac{(\|p_{m,j}\| + \|p_{m,i-1}\|)(i-j)}{2} \\ \alpha \frac{(\|p_{m,i+1}\| + \|p_{m,k}\|)(k-i)}{2} \end{bmatrix} \qquad (3)$$

where α is the azimuth angular resolution of the Velodyne LIDAR. Only the points, whose smooth arc length feature meets $Al(1) > T_1 || Al(2) > T_1$, can be selected as the candidate feature points.

3) Height Difference Feature: Based on the fact that the curbs are the road boundaries with certain height with respective to the smooth road surface, we use the height difference as an important feature to select the feature points. Let S be the set of consecutive points of $p_{m,i}$, $Zmax, Zmin$ are the maximal and minimal elevations in S, respectively. Only the points that meet $T_2 \leq Zmax - Zmin \leq T_3$ are selected as the candidate feature points.

Only the candidate feature points, whose three features above meet the requirements at the same time, will be selected as the final feature points. Fig. 3 show the features of each point in the 14^{th} scan line of Fig 1.

C. Selection of Initial Curb points

1) Distance Criterion: Based on the assumption that the curbs are the nearest obstacles to an ALV on a clear road, we use the distance criterion proposed in [3] to search the candidate curb points from all the feature points. Instead of dividing each scan line into four quadrants [3], this paper follows reference [23] to construct a polar grid map, which is divided into 720 segments evenly, and each segment is divided into 500 bins (every bin covers 0.1 meter) evenly to discrete the range component of the points. All the feature points are projected into the polar grid map, and for every segment, we only select the feature points in the nearest bin to form the 2D candidate curb point set S_C (the z coordinates of the feature points are omitted).

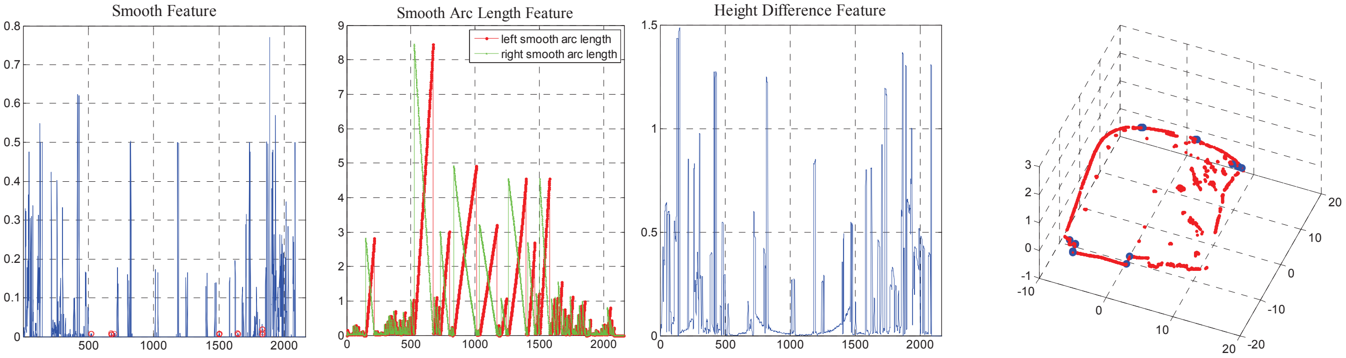

Fig. 3. Features of each point in one scan line. The blue points in the fourth image are the final feature points ($T_0 = 0.005, T_1 = 0.6, T_2 = 0.05, T_3 = 0.3$)

2) Hough Transform: This paper exploits HT to select the initial curb points from the candidate set S_C. There are at least two merits: (1) it is robust to the measurement noises when detecting straight lines; (2) the lines detected by HT could contain some global information of the distribution of the points. Though some curbs may be occluded by some objects(like the cars by the road), HT can obtain promising performance. In this step, we firstly partition these candidate curb points in S_C into a 2D grid map which covers an area of 100 by 100 meters with a resolution 0.2 meter, and then utilize HT to extract some candidate straight-line curbs as the initial seeds of GPR, which are represented by $L = \{l_0, \cdots, l_m\}, l_i = (\rho_i, \theta_i)$, where ρ_i and θi represent the distance to the original point $(0, 0)$ and the orientation of the line l_i, respectively. Considering the fact that our ALV are always driving on the road, all these candidate straight-line curbs will appear on both sides of the ALV. For line $l_i = (\rho_i, \theta_i)$, define a term d_i to represent the sign distance to the ALV.

$$d_i = v_x \cos\theta_i + v_y \sin\theta_i - \rho_i \tag{4}$$

where v_x, v_y are the x, y coordinates of the ALV in the 2D grid map, respectively. The line l_i with $(d_i \times \tan\theta_i) > 0$ is classified as the left curb, otherwise the right one. Once these candidate straight-line curbs in L are divided into two groups: the left or the right, we select the best initial curb from each group according to two constraints: (1) the angle between the orientation of the best initial curb and that of the ALV should be less than a threshold T_4; (2) we take the conservative strategy to select the nearest curb ($|d_i|$ is the minimal one) to ALV to keep safe. The candidate curb points in S_C, which locate on the best initial left or right curbs construct the set S_{il}, S_{ir} respectively, which are then used as the initial seeds of GRP. If no initial left or right curbs are detected, the successive iterative GPR will be unavailable too. The left and right initial seeds of one urban scene are illustrated in Fig 4.

D. Gaussian Process Regression based Curb Detection

"A Gaussian process (GP) is a collection of random variables, any finite number of which have a joint Gaussian distribution"[24]. Given a 2D training set $S = \{(x_i, y_i)\}_{i=1}^{n}$ of n samples, the joint Gaussian distribution of the training samples can be represented as:

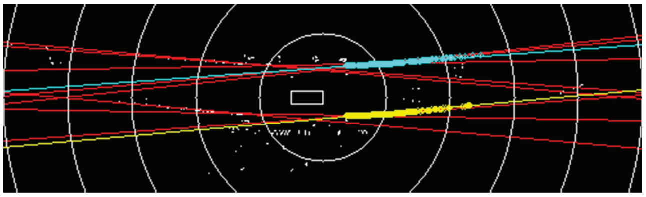

Fig. 4. The white points construct the candidate curb point set S_C, and the red lines are the candidate straight-line curbs detected by HT, where the yellow and cyan lines are the best initial right and left curbs respectively, which are chosen according to two constraints. The yellow and cyan points on the corresponding lines are the initial seeds of GRP.

$$p(Y|X) \sim N(\mu, K), \tag{5}$$

where $X = [x_1, \ldots, x_n]^T, Y = [y_1, \ldots, y_n]^T$, μ is the mean vector, and K is the covariance matrix. It is defined by a squared-exponential covariance function and a noise variance σ_n^2:

$$K(x_i, x_j) = \sigma_f^2 \exp\left(-\frac{(x_i - x_j)^2}{2l^2}\right) + \sigma_n^2 \delta_{ij}, \tag{6}$$

where δ_{ij} is a Kronecker delta, which is one iff $i = j$ and zero otherwise, l is the length-scale, and σ_f^2 is the signal covariance. $l, \sigma_f^2, \sigma_n^2$ form the hype-parameters of GPR. Based on the definition of GPR, the joint Gaussian distribution of the training outputs Y and the output y_* at the test input x_* meets:

$$\begin{bmatrix} Y \\ y_* \end{bmatrix} \sim N\left(0, \begin{bmatrix} K(X,X) & K(X,x_*) \\ K(x_*,X) & K(x_*,x_*) \end{bmatrix}\right). \tag{7}$$

Thus, the predictive equation for GPR is:

$$\begin{aligned} \bar{y_*} &= K(x_*, X)K^{-1}Y \\ V[y_*] &= K(x_*, x_*) - K(x_*, X)K^{-1}K(X, x_*) \end{aligned}, \tag{8}$$

where $\bar{y_*}$ and $V[y_*]$ are the mean and covariance of the output y_* respectively, K^{-1} is the matrix inversion of K, and $K = K(X) = \big(K(x_i, x_j)\big)_{1 \le i,j \le n}$. The k^{th} element of

$K(x_*, X) \in \mathbb{R}^{1\times n}$ is $K(x_*, x_k)$, which can be calculated using (6).

In this paper, given a candidate curb point set S_C, which may be contaminated by some noisy points, our task is to separate the read curb points S_L, S_R from S_C. Two continuous one-dimensional GPRs are exploited to built the models of the left and right curbs, respectively. However, the initial GPR assumes that all data in set S_C are curb points without any outliers, an improved iterative GPR [25], that has both the powerful approximate ability and outlier rejection ability, is adopted. The detailed procedure is shown in Algorithm 1.

Algorithm 1 Curb Detection based on Iterative GPR

```
Input:   $S_C, S_{il}, S_{ir}, T_r, l, \sigma_f, \sigma_n, t_{model}, t_{data}$
Output:  left and right curb point set $S_L, S_R$
 1: // building the left curb model
 2: $s_{new} = s_p = \emptyset$;
 3: $s_{new} = S_{il}$;
 4: while size($s_{new}$) > 0 do
 5:     $s_p = s_p \cup s_{new}$;
 6:     $model$=GPR($s_p, l, \sigma_f, \sigma_n$);
 7:     $test$=$S_C - s_p$;
 8:     $s_{new}$=eval($model, test, t_{data}, t_{model}$);
 9: end while
10: $S_L = s_p$
11: // building the right curb model
12: $s_{new} = s_p = \emptyset$;
13: $s_{new} = S_{ir}$;
14: while size($s_{new}$) > 0 do
15:     $s_p = s_p \cup s_{new}$;
16:     $model$=GPR($s_p, l, \sigma_f, \sigma_n$);
17:     $test$=$S_C - s_p - S_L$;
18:     $s_{new}$=eval($model, test, t_{data}, t_{model}$);
19: end while
20: $S_R = s_p$
```

The algorithm takes as input the candidate curb point set S_C, as well as the initial left and right curb point sets S_{il}, S_{ir}, and outputs the final left and right curb sets S_L, S_R. The algorithm mainly contains two steps: GPR with squared-exponential covariance function and new seed evaluation.

Obtaining the left and right curb model based on GPR (lines 6, 16) is the goal of the first step. The covariance matrix between the pairs of seeds s_p is calculated with (6) in this step.

The new seed evaluation is corresponding to the function $eval$ in lines 8 and 18. Equation (6) could be adopted to compute $K(x_*, x_*)$ and $K(x_*, s_p)$ for every test point (x_*, y_*) in set S_C. The mean and variance of the output y_* at the test input x_* can be predicted by (8). In this paper, the rulers (9) proposed in [25] are utilized to evaluate the test points. The parameter t_{model} is the threshold of the variance $V[y]$ of the output y_*, while t_{data} represents the normalized distance between the real output y_* and the expected mean $\bar{y}$ of the output at the test input x_*.

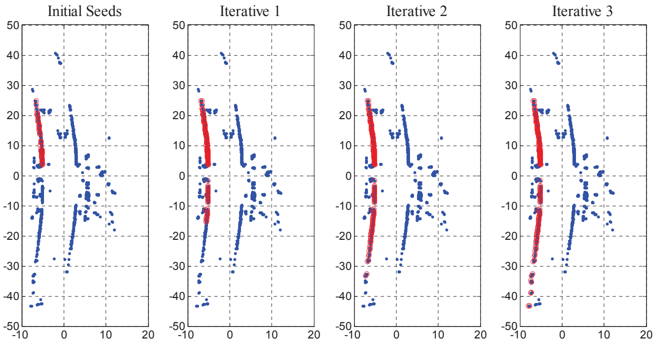

Fig. 5. The blue points are the candidate curb points in S_C, and the red circles are the accumulated seeds after every iteration when constructing the left curb model with iterative GPR ($l = 16.1003, \sigma_f^2 = 38.1416, \sigma_n^2 = 0.0039$).

$$\begin{aligned} V[y] &\le t_{model} \\ \frac{|y_* - \bar{y}|}{\sqrt{\sigma_n^2 + V[y]}} &\le t_{data} \end{aligned} \; . \qquad (9)$$

For a test point (x_*, y_*), it is selected as a new seed if and only if it meets both of the requirements in (9). Beginning with the initial left and right seeds S_{il}, S_{ir} (line 3, 13), the left and right seeds s_p are accumulated respectively until no more new seeds are selected from S_C (lines 4 - 9 and 14 - 19). Fig. 5 illustrates the iterative processes of GPR in the scene shown in Fig. 1 when constructing the left curb model.

E. Learning Hype-parameters of GPR

In order to learn the hype-parameters $\theta = (l, \sigma_f^2, \sigma_n^2)$, some manually labelled curb point sets are used as training samples $\Im = \{x^m, y^m\}_{m=1}^M$. Each curb point set is a training example of the form (x^m, y^m), where each $x^m = \{x_1^m, \ldots, x_{n_m}^m\}$ is a sequence of n_m x coordinates in curb point set m, each $y^m = \{y_1^m, \ldots, y_{n_m}^m\}$ is a corresponding sequence of the y coordinates, and n_m is the number of points in the curb point set m. As these training examples are conditionally independent, these hype-parameters θ can be computed by maximizing the pseudo log marginal likelihood [26]:

$$\begin{aligned} \textstyle\sum_{m=1}^M \log p(y^m|x^m,\theta) =& -\tfrac{1}{2}\textstyle\sum_{m=1}^M (y^m)^T K_m^{-1} y^m \\ & -\tfrac{1}{2}\log\left(\textstyle\prod_{m=1}^M |K_m|\right) \\ & -\tfrac{\log 2\pi}{2}\textstyle\sum_{m=1}^M n_m, \end{aligned} \qquad (10)$$

where K_m is the covariance matrix for the observed y^m in the curb point set m, and can be obtained with (6).

IV. EXPERIMENTAL EVALUATION AND ANALYSIS

We have evaluated the performance of the algorithm proposed in this paper on the data collected in various urban scenes with our ALL. Our ALV is a modified Toyota Land

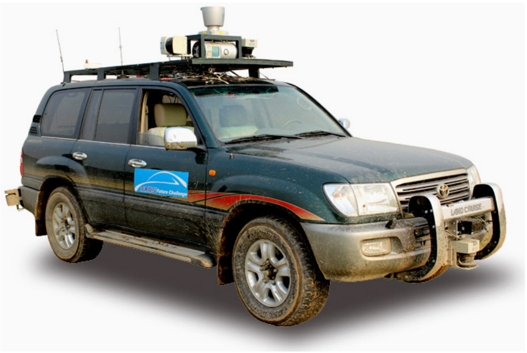

Fig. 6. Our ALV used for the experiments in urban environments. It is equipped with three SICK LMS 291, two color cameras, a Velodyne HDL-64E S2, a RIEGL laser and a NovAtel SPAN-CPT GPS-aided inertial navigation system

Cruiser, as shown in Fig. 6, equipped with a Velodyne HDL-64E LIDAR, a NovAtel SPAN-CPT GPS-aided inertial navigation system (INS) and some other sensors. The Velodyne LIDAR rotates with a speed of 10 HZ. In order to evaluate the accuracy quantitatively, we labelled 566 scans with both left and right curbs manually in various complex urban scenes, and two other algorithms are adopted to be compared with; one is the HT algorithm based on DEM to detect only straight-line curbs, and the other is our algorithm without GPR which detects the initial curb points with HT in this paper[2]. We also label some straight-line and curved curbs as training samples to learn the hype-parameters of GPR. In this paper, these parameters of GPR are set to $\sigma_f^2 = 38.1416, l = 16.1003, \sigma_n^2 = 0.0039, T_{model} = 0.49, T_{data} = 3.0$, which are learnt by maximizing the pseudo log marginal likelihood (10).

A. Quantitative Analysis

To our best knowledge, there is no public dataset for curb detection algorithm based on Velodyne LIDAR. In this paper, 2934 Velodyne scans are collected in various complex urban scenes with our ALV equipped with a Velodyne HDL-64E LIDAR, and 566 of them are labelled manually. The left and right curb points are labelled separately, but saved in the same $*.xyz$ file.

1) Runtime: In this paper, we evaluate the real-time performance of our algorithm, the algorithm without GPR and the HT algorithm based on DEM on the dataset with 2934 scans collected in various urban scenes. All these algorithms are implemented with C++ on a notebook with an Intel Core i5 CPU with 2.5 GHz using double cores and 12 GB main memory. Fig 7 shows the runtime of the three algorithms. Most of the time, with an average of 69.0ms per scan, our algorithm performs in real time, though without any optimization. One can find that our algorithm is slower than the algorithm without GPR with an average of 47.8ms per scan and the the HT algorithm based on DEM with 27.8ms per scan. Even though, our algorithm can obtain real-time performance, as the refresh rate of the scan is 100 ms.

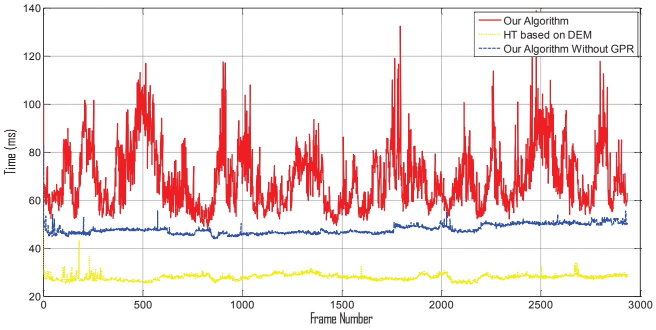

Fig. 7. Runtime of our algorithm (red) compared to that of the algorithm without GPR (green) and the HT algorithm based on DEM (yellow) for 2934 scans in various urban scenes

2) Accuracy: For these labelled scans, both the labelled left and right curb points are projected into a local grid map which covers an area of 100 by 100 meters with the resolution of 0.2 meter. Thus each cell of the grid map is classified into left curb cell, right curb cell or none. In this paper, we use each row of the grid map as the statistic unit. For example, once the cell in a row, which is detected as left curb, is also labelled as a left curb cell, we believe that this row is detected correctly for the left curb. The accuracies of left and right curb detection are detailed in Table I. One can find that, for the left curb, our algorithm can achieve an accuracy of 78.74%, which is $1.44\%, 8.44\%$ higher than the algorithm without GPR and the HT algorithm based on DEM, respectively. Meanwhile, our algorithm can also obtain best performance for right curb detection. Its accuracy is 81.96%; and the algorithm based on DEM and HT can only obtain an accuracy of 73.59%. One can also find that, as we take each scan line as a processing unit to extract feature points, the performance of left and right curb detection of the algorithm without GPR can improve by $7\%, 7.7\%$, respectively, compared with the method which detects curbs in DEM, though both of the algorithms are based on HT with the same parameters.

B. Detection Results in Some Typical Scenes

In order to evaluate the usefulness of our algorithm to our ALV intuitively, our algorithm are testified on the Velodyne scans collected in the City of Changsha, China with our ALV. For every scan, we use Point Cloud Library[27] to show the initial Velodyne data, and the corresponding curb detection results will be presented in the grid map.

Fig. 8 shows the performance of the three algorithms in four different urban scenes. The first column is the original scan data, each scan line is described by one color. The detection results of our algorithm and our algorithm without GPR are shown in the second and third columns respectively; and the fourth column is the result of the HT algorithm

[2] This algorithm only contains the steps in Subsections III-A, III-B, III-C. The results of curb detection are obtained by HT, essentially, and the results of this algorithm are also the initial seeds of GPR in our algorithm. The difference between this algorithm and the algorithm based on DEM and HT is the way how to get candidate curb points.

TABLE I

CONFUSION MATRIX OF THE THREE ALGORITHMS

	Our Algorithm		Our Algorithm Without GPR		HT based on DEM	
	Left Curb	None	Left Curb	None	Left Curb	None
Left Curb	77737	49900	70003	57634	48265	79372
None	10399	145530	6729	149200	4841	151088
Accuracy	78.74%		77.30%		70.30%	
	Right Curb	None	Right Curb	None	Right Curb	None
Right Curb	53347	42150	48690	46807	29483	66014
None	9004	179065	6238	181831	8874	179196
Accuracy	81.96%		81.29%		73.59%	

based on DEM. The left and right curb points are represented by cyan and yellow colors, and the white points are some candidate curb points. From Fig. 8, one can find that our algorithm outperforms the algorithm without GPR and the HT algorithm based on DEM, especially for the curved curb (e.g. the first, second and third rows). As these scenes contain the curved curbs, the curb detection algorithm based on HT can only obtain a part of the real curb points (like the third and fourth columns). However, our algorithm with iterative GPR can utilize small part of the curb points as initial seeds to detect the whole curb points iteratively. The second, third and fourth rows also show that our curb detection algorithm can obtain curb points up to 50 meters away, even in some environments with curved curbs, while the HT algorithm based on DEM can only get curb points at 30 meters, even for straight-line curbs.

V. CONCLUSIONS

This paper presents a novel Velodyne-based curb detection algorithm for an ALV. Each scan line of LIDAR is directly taken as a processing unit, which makes our algorithm detect the curb up to 50 meters away. Firstly, the smooth feature, smooth arc length feature and height difference feature are calculated to extract the feature points from each scan line, then distance criterion is exploited to select the candidate curb points. In order to detect both the straight-line and curved curbs, iterative GPR, whose initial seeds are provided by HT, is utilized to build the curb models. In order to evaluate our curb detection algorithm quantitatively, 566 Velodyne scans collected by our ALV in various urban environments are manually labelled. Both the quantitative and qualitative experiments show promising results.

VI. ACKNOWLEDGEMENT

Thank my colleagues (Defeng Sun, Xiangfeng Zeng, Wei Zhao and Peidong Wang) who help me label some Velodyne scans manually.

REFERENCES

[1] Q. Zhu, L. Chen, Q. Li, M. Li, A. Nuchter, and J. Wang, "3d lidar point cloud based intersection recognition for autonomous driving," in *IEEE Intelligent Vehicles Symposium*, 2012, pp. 456–461.

[2] A. H. Hata, D. habermann, F. S. Osorio, and D. F. Wolf, "Road gromentry classifiaction using ann," in *IEEE Intelligent Vehicles Symposium*, 2014, pp. 1319–1324.

[3] A. H. Hata, F. S. Osorio, and D. F. Wolf, "Robust curb detection and vehicle localization in urban environments," in *IEEE Intelligent Vehicles Symposium*, 2014, pp. 1257–1262.

[4] B. Qin, Z. J. Chong, T. Bandyopadhyay, M. H. Ang, E. Frazzoli, and D. Rus, "Curb-intersection feature based monte carlo localization on urban roads," in *International Conference and Robotics and Automation*, 2012, pp. 2640–2646.

[5] G. Zhao and J. Yuan, "Curb detection and tracking using 3d-lidar scanner," in *International Conference on Information Processing*, 2012, pp. 437–440.

[6] J. Siegemund, U. Franke, and W. Forstner, "A temporal filter approach for detection and reconstruction of curbs and road surfaces based on conditional random fields," in *IEEE Intelligent Vehicles Symposium*, 2011, pp. 637–642.

[7] F. Oniga and S. Nedevschi, "Polynomial curb detection based on dense stereovison for driving assistance," in *IEEE International Conference on Intelligent Transportation Systems*, 2010, pp. 1110 –1115.

[8] F. Oniga, S. Nedevschi, and M. M. Meinecke, "Curb detection based on a multi-frame persistence map for urban driving scenarios," in *IEEE International Conference on Intelligent Transportation Systems*, 2008, pp. 67–72.

[9] V. Pradeep, G. Medioni, and J. Weiland, "Piecewise planar modeling for step detection using stereo vision," in *Workshop on computer vision applications for the visually impaired*, 2008.

[10] M. Kellner, M. E. Bouzouraa, and U. Hofmann, "Road curb detection based on different elevation mapping techniques," in *IEEE Intelligent Vehicles Symposium*, 2014, pp. 1217–1224.

[11] J. Stuckler, H. Schulz, and S. Behnke, "In-lane localization in road networks using curbs detected in omnidirectional height images," in *Proceedings of Robotics*, 151-154 2008.

[12] Z. Liu, D. Liu, T. Chen, and C. Wei, "Curb detection using 2d range data in a campus environments," in *International Conference on Image and Graphics*, 2013, pp. 291–296.

[13] A. Hervieu and B. Soheilian, "Road side detection and reconstruction using lidar sensor," in *IEEE Intelligent Vehicles Symposium*, 2013, pp. 1247–1252.

[14] W. Zhang, "Lidar-based road and road-edge detection," in *IEEE Intelligent Vehicles Symposium*, 2010, pp. 845–848.

[15] W. S. Wijesoma, K. R. S. Kodagoda, and A. P. Balasuriya, "Road boundary detection and tracking using ladar sensing," *IEEE Transaction on Robotics and Automation*, vol. 20, no. 3, pp. 456–464, 2004.

[16] K. R. S. Kodagoda, S. S. Ge, W. S. Wijesoma, and A. P. Balasuriya, "Immpdaf approach for road-boundary tracking," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 2, pp. 478–486, 2007.

[17] M. Kellner, U. Holfmann, M. E. Bouzouraa, H. Kasper, and S. Neumaier, "Laserscanner based road curb feature detection and efficient mapping using local curb descriptors," in *IEEE International Conference on Intelligent Transportation Systems*, 2014, pp. 2602–2608.

[18] K. Peterson, J. Ziglar, and P. E. Rybski, "Fast feature detection and stochastic parameter estmation onf raod shape using multiple lidar," in *IEEE International Conference on Intelligent Robots and Systems*, 2008, pp. 612–619.

[19] K. R. S. Kodagoda, W. S. Wijesoma, and A. P. Balasuriya, "Cute: Curb tracking and estimation," *IEEE Transactions on Control Systems Technology*, vol. 14, no. 5, pp. 951–957, 2006.

[20] J. Tan, J. Li, X. An, and H. He, "Robust curb detection with fusion of 3d-lidar and camera data," *Sensors*, pp. 9046–9073, 2014.

[21] X. Yuan, C. X. Zhao, and H. F. Zhang, "Road detection and corner ex-

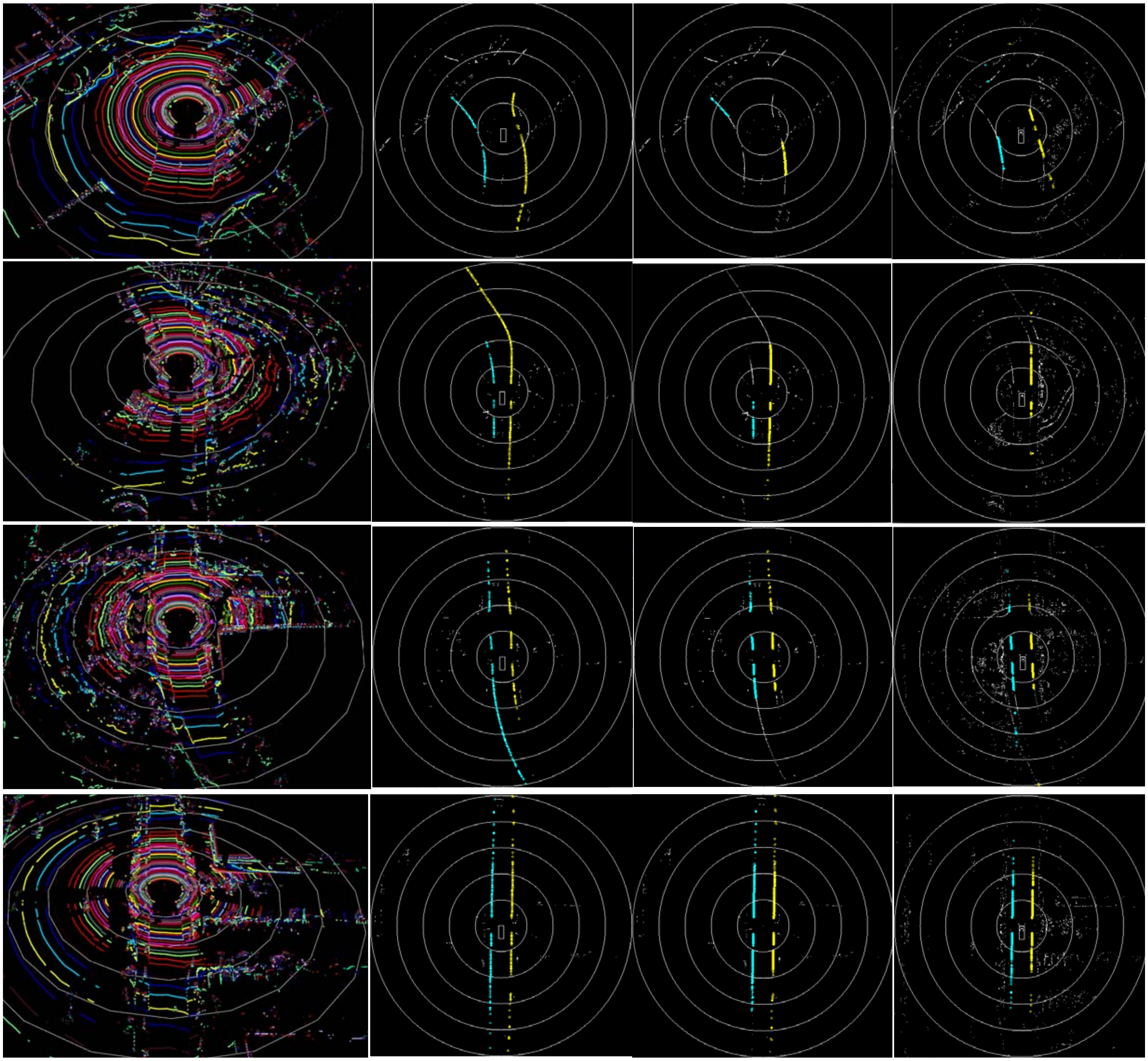

Fig. 8. The detection results of three different curb detection algorithms in four complex urban scenes. The first column is the original scan data, and each color represents one scan line. The second, third and fourth columns are the curb detection results of our algorithm, our algorthim without GPR and the algorithm based on DEM and HT, respectively. The cyan pixels are left curb points, while the yellows are right ones. The white pixels represent the candidate curb points. Each white circle in this figure represent 10 meters

traction using high definition lidar," *Information Technology Journal*, vol. 9, no. 5, pp. 1022–1030, 2010.

[22] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time," in *Robotics: Science and Systems Conference*, 2014, pp. 1–8.

[23] M. Himmelsbach, F. Hundelshausen, and H. J. Wuensche, "Fast segmentation of 3d point clouds for ground vehicles," in *IEEE Intelligent Vehicles Symposium*, 2010, pp. 560–565.

[24] C. E. Rasmussen and C. K. Williams, *Gaussian Processes for Machine Learning*. England: the MIT press, 2006.

[25] B. Douillard, J. Underwood, N. Kuntz, V. Vlaskine, A. Quadros, P. Morton, and A. Frenkel, "On the segmentation of 3d lidar point clouds," in *International Conference on Robotics and Automation*, 2011, pp. 2798–2805.

[26] T. Chen, B. Dai, R. Wang, and D. Liu, "Gaussian-process-based real-time ground segmentation for autonomous land vehicle," *Journal of Intelligent and Robotic systems*, vol. 76, no. 3, pp. 563–582, 2014.

[27] R. Rusu and S. Cousins, "3d is here: point cloud library (pcl)," in *International Conference on Robotics and Automation (ICRA)*, 2011, pp. 1–4.

View publication stats