3D LIDAR-Based Intersection Recognition and Road Boundary Detection Method for Unmanned Ground Vehicle

Conference Paper · September 2015

DOI: 10.1109/ITSC.2015.88

5 authors, including:

Yihuan Zhang
Tsinghua Automotive Research Institute (Suzhou)
15 PUBLICATIONS **50** CITATIONS

SEE PROFILE

Jun Wang
Tongji University
79 PUBLICATIONS **713** CITATIONS

SEE PROFILE

Wang Liang
Tongji University
6 PUBLICATIONS **19** CITATIONS

SEE PROFILE

2015 IEEE 18th International Conference on Intelligent Transportation Systems

3D LIDAR-based Intersection Recognition and Road Boundary Detection Method for Unmanned Ground Vehicle

Yihuan Zhang[1], Jun Wang[1*], Xiaonian Wang[2], Chaocheng Li[3] and Liang Wang[4]

***Abstract*— Environment perception is an essential component in autonomous driving technology. Curb is one of the most prominent features on an urban road, which defines the boundary of a road surface. An intersection is a junction of two or more roads, appearing where no curb exists. Intersection recognition and curb detection are significant in environment perception which ensure the performance of Unmanned Ground Vehicles (UGVs). In this paper, a novel double layer beam model is proposed to recognise the intersection shape and classify the road type in front of the UGV, and then based on the current road type and spatial features, a real-time road boundary detection algorithm is proposed to extract the curb position. The performance of the proposed method is verified through extensive experiments with a UGV autonomously driving on campus roads. The experimental results demonstrate the accurate and robust performance of the proposed algorithm.**

I. Introduction

Autonomous driving technology is rapidly growing to meet the needs of road safety and efficiency of transportation. UGVs can be used for many applications where it may be inconvenient, dangerous, or impossible to have a human operator on site. Environment perception is one fundamental requirement for vehicle autonomy, and curb detection is the foremost problem in perception system.

In the past few years, researchers have applied many different types of sensors including video sensors, millimetre wave radars (MMWR) and laser scanner sensors to curb detection. In [1], a stereo vision camera was used to find the curb and the Canny algorithm was employed based on a multi-frame persistence map. In [2], the curbs were found by using a stereo vision camera as well, the conditional random field (CRF) algorithm was proposed to propagate the curb detection result. In [3], the MMWR was applied for road boundary detection, it considered both road and off-road scatter components' difference and derived the fisher information matrix to evaluate the achievable accuracy. In [4], an MMWR sensor and a camera sensor were used to detect the lane and pavement boundaries. In [5], a scanning two-dimensional laser measurement system was used to detect and track the road boundary and the method was based on an extended Kalman Filter to enhance the efficiency and accuracy. The drawback of all the above methods is that a lot of false positive detections are produced due to the limited amount of sensing data. In order to obtain the overall information of the surrounding environment, high accuracy 3D-LIDAR sensor was used to percept the environment in [6]. A 3D-LIDAR sensor was used to obtain large amount of data of surrounding environment, the result showed that the method proposed was accurate and robust in most scenarios.

In some scenarios such as intersections which have no curb. In order to detect the curb and represent the road accurately, many researches focused on the intersection detection. Many existing intersection detection algorithms are proposed in the field of remote sensing. In [7], intersections were detected based on aerial images which were not suitable for UGVs. For UGVs, cameras were widely used in the past few years [8], [9]. However, the vision-based methods relied on the light condition and were nearly impossible to detect the intersection when the light condition was poor. In [10], the LIDAR sensor was used to detect the two edges of the road and the intersections, which was not effected by light condition. In [11] and [12], the 3D LIDAR sensor were used to classify the intersection type based on a beam model. The result demonstrated the accuracy of the detection method. Those methods only concern about the recognition of intersection.

In this paper, the Velodyne HDL-32E 3D-LIDAR sensor is mounted on the roof of our UGV, which can produce accurate and sparse point cloud. The proposed method in this paper is focusing on the intersection recognition and road boundary detection in real time. The road boundary are defined as the curbs and the road border which has a relatively high elevation from the road surface. First the raw data from the sensor is preprocessed to eliminate ground points and obstacles on road, then based on a dual beam model the road segments are separated. According to road segments and spatial feature, the curb points are obtained in each segment of the road. A parabola model is used to fit the curb points and border points of each segment to represent the road.

The remainder of the paper is structured as follows. In Section II, the intersection recognition method is detailed along with the data preprocessing steps. In Section III, the curb detection method is proposed and the parabola model is introduced. The experiment results are presented in Section IV followed by the conclusion in Section V.

II. Intersection Recognition

This section introduces the intersection recognition method based on a double layer beam model. First, based on the elevation information, the road surface points are filtered of each frame. Then a bottom-layer beam model is built to determine the initial beam angle and length. A top-

Yihuan Zhang, Jun Wang, Xiaonian Wang, Chaocheng Li and Liang Wang are with the Department of Control Science and Engineering, Tongji University, Shanghai 201804, P. R. China.

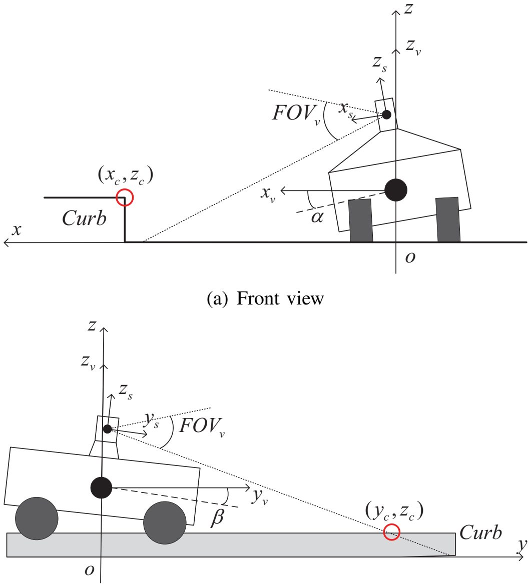
*Corresponding author junwang@tongji.edu.cn

978-1-4673-6596-3/15 $31.00 © 2015 IEEE
DOI 10.1109/ITSC.2015.88

layer beam model is used to extract the feature of the road segment based on the classification rules.

A. Data preprocessing

The HDL-32E LIDAR sensor is small, lightweight, ruggedly built and features up to 32 lasers across a 40° vertical field of view. The 32 lasers of the sensor are aligned from +10° to -30° to provide an unmatched vertical field of view, and its rotating head design delivers a 360° horizontal field of view natively. The HDL-32E generates a point cloud of 700,000 points per second with a range of 70 m and typical accuracy of 2 cm [13].



(a) Front view

(b) Side view

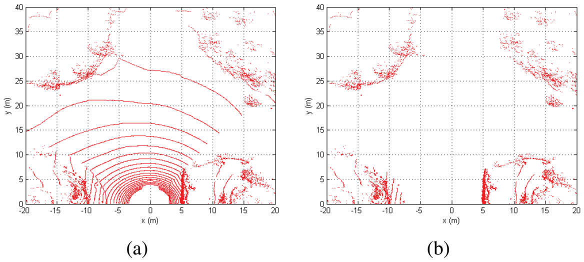
Fig. 1. Coordinate system and curb description

In this paper, the velodyne sensor is mounted on the top of the UGV. The raw data is in a 3-D polar coordinate $(\rho_i, \theta_i, \gamma_i)$, where ρ_i represents the spatial distance from the point to the sensor, θ_i and γ_i are the horizontal and vertical angle with respect to the sensor coordinate system. The cartesian coordinate system we define is shown in Fig. 1. The raw data in cartesian coordinate is represented by a set of points $\mathbf{R}_i = (x_i, y_i, z_i)$ and can be calculated by the following Equations (1) and (2):

$$\begin{cases} x_{i,s} = \rho_i \cos(\gamma_i) \sin(\theta_i) \\ y_{i,s} = \rho_i \cos(\gamma_i) \cos(\theta_i) \\ z_{i,s} = \rho_i \sin(\gamma_i) \end{cases} \tag{1}$$

$$[x_i, y_i, z_i] = [x_{i,s}, y_{i,s}, z_{i,s} - z_0]\boldsymbol{T_r} \tag{2}$$

where $\boldsymbol{T_r}$ is the transfer matrix and z_0 is the height from the sensor's centre of gravity to the ground, α and β are the roll and pitch angle of the vehicle provided by the inertial measurement unit. Based on the elevation feature and the continuity of road surface, the road surface points are eliminated in order to set up beam model. Fig. 2 shows the performance of the data preprocessing method of a single frame.



(a)

(b)

Fig. 2. (a) raw data distribution under the vertical view; (b) filtered data distribution under the vertical view.

B. Double layer beam model

Beam model is an approximate physical model of range finders and widely used in robotics [14]. It is a sequence of beams with common initial point where the range finders are located and evenly spaced with angle resolution $\delta = 2\pi/n$. In this paper, a double layer beam models are built to recognise the intersection.

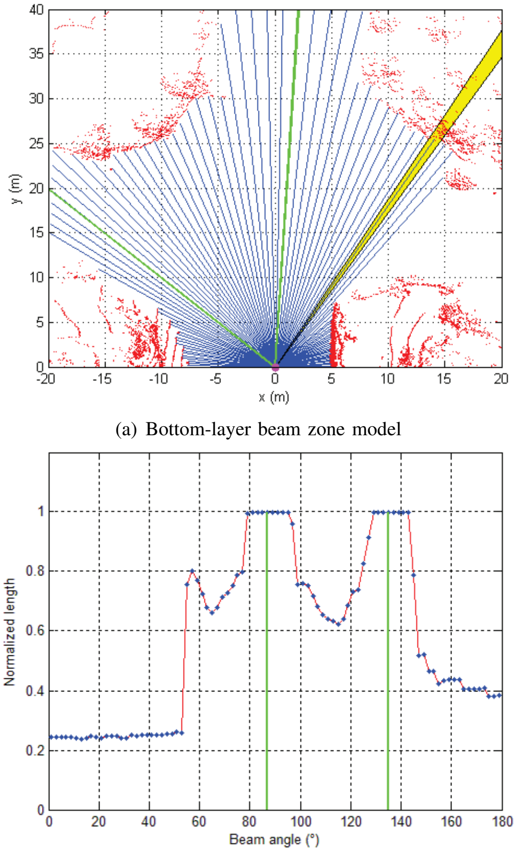
1) Bottom-layer beam model: The bottom-layer beam model is used to determine the initial beam point of the top-layer beam model. By setting up the initial point at the vehicle position, the initial angle is determined based on the peak-finding method. The filtered points in one frame are defined as: $\mathbf{P}_i = (x_i, y_i, z_i)$, in order to consider all the points, we modify the beam model by changing the beam line into a beam zone. The improved beam model is denoted as follows:

$$\mathbf{Z}_k = \left\{ \frac{(k-1)\cdot\pi}{n} < \arctan\left(\frac{y - y_{ini}}{x - x_{ini}}\right) \leq \frac{k\cdot\pi}{n}, \right.$$
$$\left. -20 \leq x \leq 20, 0 \leq y \leq 40, k = 1, 2, \cdots, n \right\} \tag{3}$$
$$d_k = \left\{ \min_{\mathbf{P}_j \in \mathbf{Z}_k} \sqrt{x_j^2 + y_j^2}, k = 1, 2, \cdots, n \right\}$$

where $\mathbf{Z}_k$ means the k_{th} beam zone area, d_k presents the shortest distance among the points $\mathbf{P}_j$ in beam zone k and x_{ini}, y_{ini} means the launching point of the beam model. The advantage of the improved beam model is that it considers all the filtered points in one frame and finds the shortest distance of each beam zone.

As shown in Fig. 3(a). The purple point is the initial point of the model. The yellow area illustrates one beam zone. The blue lines are in the middle of each beam zone and the length shows the shortest length in the zone. The green line represents the initial angle for the top-layer beam model which is found by rule-based peak-finding method. The result of the peak-finding method is shown in Fig. 3(b), according to the Equation (3), the normalized beam length are calculated as follows:

$$D_k = \frac{d_k}{\max_{x,y \in \mathbf{Z}_k} \sqrt{x^2 + y^2}} \tag{4}$$



(a) Bottom-layer beam zone model

(b) Normalized beam length and initial angle

Fig. 3. Bottom-layer beam model

where D_k represents the normalized length of each beam zone which are shown in blue dots in Fig. 3(b). The peaks $\{(\Theta_{b,i}, d_{b,i}), i = 1, 2, \cdots, n\}$ are shown in green in Fig. 3, which are initial angles for the top-layer beam model.

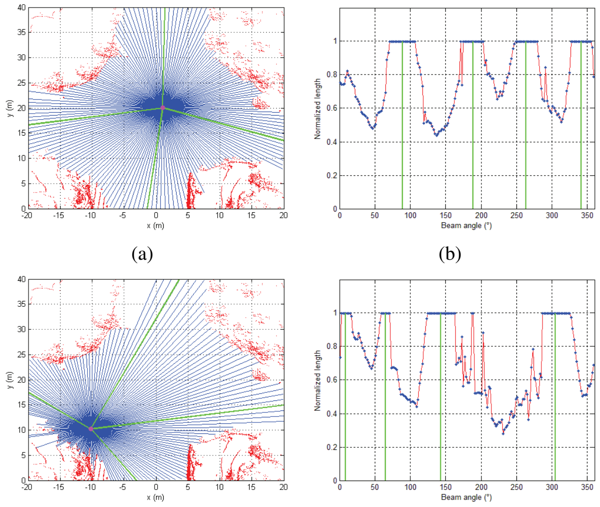
2) Top-layer beam model: The top-layer beam model is built to recognise the road shape, especially the intersection. Based on the initial angles provided by the bottom-layer, top-layer beam model is built to find the number of branches in one frame. The initial beam points are calculated as follows:

$$\begin{cases} x_{ini} = d_{b,i} \cdot \omega_d \cdot \cos \Theta_{b,i} \\ y_{ini} = d_{b,i} \cdot \omega_d \cdot \sin \Theta_{b,i} \\ \omega_d = \omega_{ref} + \frac{v}{v_{max}} \end{cases} \tag{5}$$

where $i = 1, 2, \cdots, n$, v represents the current velocity and v_{max} is the speed limitation of the road, in the case mentioned above, there are two initial beam points. Then two separate beam zone models are built and the results are shown in Fig. 4. The peaks $\{(\Theta_{t,i}, d_{t,i}), i = 1, 2, \cdots, n\}$ in each model are found by the peak-finding method where $n = 4$ in Fig. 4. The initial beam point in Fig. 4(a) is chosen to be the segmentation center, so the road segmentation are represented as follows:

$$\mathbf{S}_k = \Big\{ \Theta_{t,k-1} < \arctan\left(\frac{y - y_{ini}}{x - x_{ini}}\right) \leq \Theta_{t,k}, \\ k = 1, 2, \cdots, n, \Theta_{t,0} = \Theta_{t,n} \Big\} \tag{6}$$

The road border is represented as the set of the beam points with the normalized length less than 1, the set can be presented as: $\mathbf{B} = \{(\Theta_i, d_i), i = 1, 2, \cdots, m\}$, where m is the number of beam points.



(a)

(b)

(c)

(d)

Fig. 4. Top-layer beam model

In this paper, we mainly focus on the $+$-shaped and T-shaped intersections. Based on the branches obtained above, the intersection is simply classified as a $+$-shaped intersection. In most scenarios, the number of branch is 2 which represents a normal road and a T-shaped intersection has 3 branches. The experiment results are shown in Section IV.

III. Road Boundary Detection

In Section II, the road shape is classified and the segmentations are obtained. In order to detect the road boundary, the point cloud filtering algorithm is proposed to prepare for the road boundary detection algorithm.

A. Point cloud filtering

As the Velodyne sensor is introduced in Section II. Based on the spatial feature on the road, algorithm 1 is proposed to filter the raw data. In the nearby area of the UGV, the sensor data is intensive and able to cover the nearby curb and road surface. On the contrary, the data in distant area is sparse and is unable to detect the intact curb and road surface.

The result of algorithm 1 is shown in Fig. 5. The blue points represent the filtered data, and the red points are border points outside the road. According to the segmentation obtained before, the filtered points are divided into four zones: $\mathbf{Q}_k = \{\mathbf{P} \subseteq \mathbf{S}_k, k = 1, 2, 3, 4\}$. After the filter

Algorithm 1 Raw data filtering algorithm

```
Require:
    The raw data set, $\mathbf{R}$;
    The euclidean distance threshold of $x$-$y$ plane, $\delta_{xy}$;
    The vertical distance threshold, $\delta_z$;
    The quantitative threshold, $N$;
Ensure:
    The filtered data set, $\mathbf{P}$;
 1: Separate the raw data set $\mathbf{R}$ into each laser $\mathbf{R}_i$, where
    $i = 1, 2, \cdots, 32$ and initialize $i = 0$;
 2: $i = i + 1$;
 3: adjust the $\delta_{xy}$, $\delta_z$, and $N$ according to $i$ and previous
    parameters;
 4: For each point $\mathbf{R}_{i,j} = (R_{i,j,x}, R_{i,j,y}, R_{i,j,z})$ in $\mathbf{R}_i$,
    calculate the euclidean distance of $x$-$y$ plane $p_{dist} =$
    $\sqrt{(R_{i,j,x} - R_{i,j+1,x})^2 + (R_{i,j,y} - R_{i,j+1,y})^2}$ and the
    vertical distance $\Delta z = |R_{i,j,z} - R_{i,j+1,z}|$ between two
    adjacent points;
 5: if $\Delta z \leq \delta_z$ & $p_{dist} \leq \delta_{xy}$ then
 6:     if $Jump\_flag$ is set then
 7:         Put $\mathbf{R}_{i,j}$ into the list;
 8:         if the length of the list $\geq N$ then
 9:             Label the points in the list;
10:             Reset the $Jump\_flag$;
11:             Clear the list;
12:         end if
13:     else
14:         Label the recent point $\mathbf{R}_{i,j}$
15:     end if
16: else
17:     Set the $Jump\_flag$;
18:     Clear the list;
19:     Put $\mathbf{R}_{i,j}$ into the list;
20: end if
21: Put the labeled points into $\mathbf{P}_i$;
22: While $i \leq 32$, repeat 2 - 21
23: return $\mathbf{P}$;
```

algorithm the relatively smooth data sequences of each segmentation are obtained, which have more noticeable features between the road surface and the curbs.

B. Road boundary extraction

In this paper, the road boundary is defined as a combination of curbs and borders of the road. There are many spatial features of curb points that differ from the road surface points. First, the curb height is uniform in most urban areas and it is often 10 to 15 cm higher than the road surface. Second, the elevation changes sharply in the z direction of the cartesian coordinate. Third, the road surface points are smooth and continuous, and the curb always appears at the two sides of the road in most cases. Based on the intersection recognised before, the curb detection algorithm is proposed to detect the curb points. The algorithm 2 is proposed to detect the curb and road boundary.

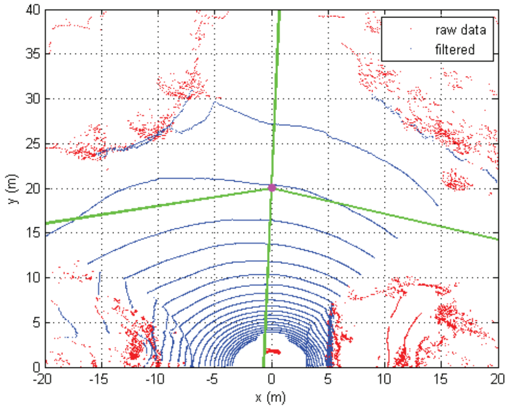


Fig. 5. Raw data filtering result

The result of the proposed method is shown in Fig. 6. Curb points $\mathbf{Q}_k$ are shown in purple square in each zone. The road boundary in each zone is represented as a green curve after the parabola fitting. In this scenario, it is unable to detect the curbs in two distant zones so the border points are fitted by parabola model to represent the road boundary.

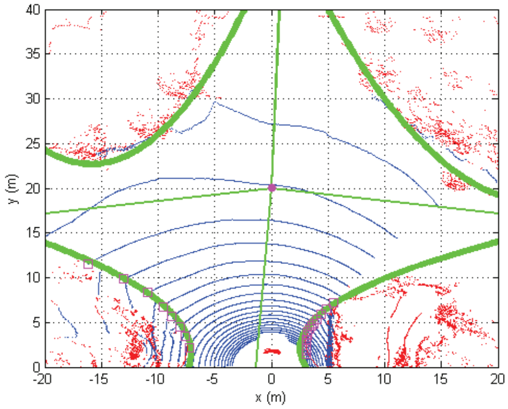


Fig. 6. Road boundary detection algorithm result

IV. Experiment results

To evaluate the proposed method, the experiments are carried out both online and offline. Furthermore, we compare the proposed method with two state-of-the-art methods which are Haar Wavelet Transformation [15] and Hough Transformation [16]. Both methods have their own advantages and disadvantages which are shown in the experiment results.

A. Dataset experiment

In order to compare with the two different methods, a dataset recorded by our UGV is used to test these methods in MATLAB. The dataset consists of the vehicle velocity, heading orientation, pitch and roll angle, along with the raw data from Velodyne sensor of each frame. The three methods are evaluated based on a sequence of 100 frames (approximately 80m) in the dataset. Parameters of the curb detection and tracking algorithm are defined in Table I. As

Algorithm 2 Road boundary detection algorithm

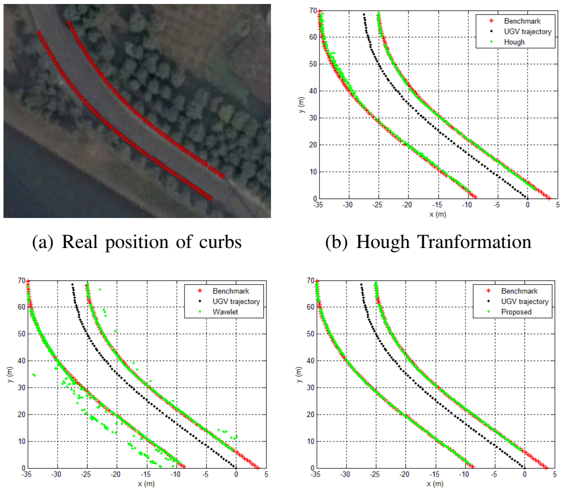
```
Require:
    The Segmented data set, $\mathbf{Q}$;
    The euclidean distance threshold of $x$-$y$ plane, $\xi_{xy}$;
    The vertical distance threshold, $\xi_z$;
    The width of a sliding window, $wid$;
    The offset of a sliding window, $N_{off}$;
    The segmentation number, $n$;
Ensure:
    The segmented road boundary set, $\mathbf{Bo}$;
 1: Separate $\mathbf{Q}_k$ by each laser $\mathbf{Q}_{k,i}$, where $i = 1, 2, \cdots, 32$;
 2: for $k = 1$ to $n$ do
 3:   for $i = 1$ to 32 do
 4:     adjust the $\xi_{xy}$, $\xi_z$, $wid$ and $N_{off}$ according to $i$ and previous parameters;
 5:     for each $j$ in $\mathbf{Q}_{k,i}$ do
 6:       Put from $\mathbf{Q}_{k,i,j}$ to $\mathbf{Q}_{k,i,j+wid}$ into sliding window $W_j$;
 7:       if $(\max_z(W_{j,z}) - \min_z(W_{j,z})) \geq \xi_z$ then
 8:         for $m$ in $W_j$ do
 9:           calculate the $p_{dist}$;
10:           if $p_{dist} \geq \xi_{xy}$ then
11:             Label $\mathbf{Q}_{k,i,j}$ and jump to 3;
12:           else
13:             Calculate the $\Delta z$ between the adjacent points and label $\mathbf{Q}_{k,i,j}$ with the maximum $\Delta z$ and jump to 3;
14:           end if
15:         end for
16:       else
17:         $i = i + N_{off}$ and jump to 6;
18:       end if
19:     end for
20:     Put the labeled points into $\mathbf{Q}_k$;
21:   end for
22:   Fitting $\mathbf{Q}_k$ using parabola model and compare the distance with beam length;
23:   Put the closer points into $\mathbf{Bo}_k$
24: end for
25: return $\mathbf{Bo}$;
```

shown in Fig. 7(a), the curb points in Google Earth are labeled in red and the curbs are modified using the method of Open Street Map (OSM) program in [17]. In [18], the bertha benz self-driving car uses the OSM data as a static curb reference to plan the global trajectory. The dataset experiment results of three different detection methods are shown in Fig. 7. The result in Fig. 7(b) shows that the Hough Transformation method has difficulty in handling high curvature roads, the curb points scatteres at the curve part of the road. In Fig. 7(c), there are many false positive detections. Although the method can provide relatively accurate curb points, it can not be used in our UGV without noise filtering. As shown in Fig. 7(d), with the pre-processing of the raw data and a dynamic threshold adjustment, the sparse curb points are relatively accurate. In addition, a parabola model is used to fit the sparse curb points to obtain the dense curb points which is shown in green.

TABLE I

ESSENTIAL PARAMETER INITIALIZATION

Parameters	Initialization	Tolerance($\pm$)
δ_{xy}	0.3 m	0.1 m
δ_z	0.08 m	0.03 m
N	20	10
ξ_{xy}	0.2 m	0.08 m
ξ_z	0.05 m	0.02 m
wid	30	10
N_{off}	10	5



(a) Real position of curbs

(b) Hough Tranformation

(c) Haar Wavelet Tranformation

(d) Proposed Method

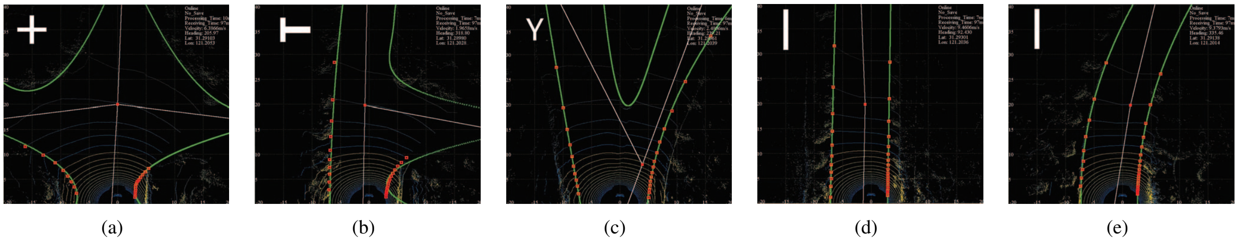
Fig. 7. (a) shows the global map in Google Earth, and the red nails represent real curb position; (b), (c), (d) show the curb distribution using different methods compared with the real curb position.

The performance indexes are shown in Table II, ε is set to 0.1 m. Though the short average period of Haar Wavelet Transformation method, its precision rate is lower than the proposed method. Hough Transformation method has a relatively high precision rate but it is too time-consuming for real-time operation. The proposed method has the highest precision rate and acceptable processing period among three methods in the dataset experiment. When it comes to an intersection, the Haar and Hough method are unable to detect the curb, so in the real-time experiment, we only test the proposed method.

TABLE II

PERFORMANCE INDEX OF THREE METHODS

Methods	Average period	Precision rate (ε)	Precision rate (3ε)
Proposed	25.3 ms	96.88%	100.00%
Hough	73.2 ms	83.22%	98.76%
Wavelet	16.4 ms	78.91%	86.79%



(a) (b) (c) (d) (e)

Fig. 8. (a) $+$-shaped intersection scenario, (b) T-shaped intersection scenario, (c) Y-shaped intersection scenario, (d) straight road scenario, (e) curve road scenario. All scenarios are intercepted from the PXI controller in realtime.

B. Real-time experiment

After the comparison of the three methods, the proposed algorithm is tested on the UGV platform which uses a PXI as the core controller. The NI PXI-8109 is a high-performance intel core i7-620M processor-based embedded controller for PXI systems. With the 2.66 GHz base frequency, PXI is able to process the Velodyne sensor data in realtime. The proposed method is realised in Visual Studio based on Windows-7 Operating System. Real-time results of some road scenarios are shown in Fig. 8. The white lines launching from the red square represents the road segmentations. In each segmentation, the red hollow squares are the curb points extracted from the road and the green curve is the road boundary points. In Fig. 8(a) and 8(b), the intersection is near the vehicle and the curb points can not be found in the distant area because of the limitation of the sensor. Road border position presents a relatively accurate results of road boundary when there are no curbs. Experiments demonstrate that based on the recognition of intersection, the performance of road boundary detection method is improved observably.

V. Conclusion and Future Work

This paper develops a real-time intersection recognition and road boundary detection algorithm using a Velodyne 3D-LIDAR sensor. Based on the double layer beam model, the intersection shape is recognized and the road is segmented. Then the raw data are filtered and the curbs are detected using dynamic sliding window method based on spatial features. Also, the curbs are combined with the border points to cover all segmentations on road and parabola model is used to fit the road boundary points in each segmented zone. Experiments on dataset demonstrate the accuracy and robustness of the proposed method. The real-time experiment shows the proposed method is time-efficient and accurate for curb detection and tracking.

The correctness of intersection recognition in some abnormal scenarios such as a Y-shaped intersection is a challenge to solve in the near future. The machine learning method can be implied to train the big data gathered from the UGV.

Acknowledgment

This work is supported by the National Science Foundation of China under Grant No.61473209.

References

[1] F. Oniga, S. Nedevschi, and M. M. Meinecke, "Curb detection based on a multi-frame persistence map for urban driving scenarios," in *the 11th International IEEE Conference on Intelligent Transportation Systems. (ITSC)*. IEEE, 2008, pp. 67–72.

[2] J. Siegemund, D. Pfeiffer, U. Franke, and W. Forstner, "Curb reconstruction using conditional random fields," in *Proceedings of the 2010 IEEE Intelligent Vehicles Symposium. (IV)*. IEEE, 2010, pp. 203–210.

[3] M. Nikolova and A. Hero, "Segmentation of a road from a vehicle-mounted radar and accuracy of the estimation," in *Proceedings of the 2000 IEEE Intelligent Vehicles Symposium. (IV)*. IEEE, 2000, pp. 284–289.

[4] B. Ma, S. Lakshmanan, and A. O. Hero, "Simultaneous detection of lane and pavement boundaries using model-based multisensor fusion," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 3, pp. 135–147, 2000.

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

[6] G. Zhao and J. Yuan, "Curb detection and tracking using 3d-lidar scanner," in *the 19th IEEE International Conference on Image Processing. (ICIP)*. IEEE, 2012, pp. 437–440.

[7] J. Hu, A. Razdan, J. C. Femiani, M. Cui, and P. Wonka, "Road network extraction and intersection detection from aerial images by tracking road footprints," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 12, pp. 4144–4157, 2007.

[8] C. Rasmussen, "Road shape classification for detecting and negotiating intersections," in *Proceeding of Intelligent Vehicles Symposium (IV)*. IEEE, 2003, pp. 422–427.

[9] M. S. Kurdziel, *A monocular color vision system for road intersection detection*. ProQuest, 2008.

[10] K. S. Kodagoda, W. S. Wijesoma, and A. P. Balasuriya, "Road curb and intersection detection using a 2d lms," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2002.*, vol. 1. IEEE, 2002, pp. 19–24.

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

[12] C. Tongtong, D. Bin, L. Daxue, and L. Zhao, "Lidar-based long range road intersection detection," in *Image and Graphics (ICIG), 2011 Sixth International Conference on*. IEEE, 2011, pp. 754–759.

[13] "Velodyne hdl 32-e lidar," http://www.velodynelidar.com/lidar/hdlproducts/hdl32e.aspx.

[14] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.

[15] K. Peterson, J. Ziglar, and P. E. Rybski, "Fast feature detection and stochastic parameter estimation of road shape using multiple lidar," in *Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems. (IROS)*. IEEE, 2008, pp. 612–619.

[16] Y. Kang, C. Roh, S. Suh, and B. Song, "A lidar-based decision-making method for road boundary detection using multiple kalman filters," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 11, pp. 4360–4368, 2012.

[17] F. Ramm, J. Topf, and S. Chilton, *OpenStreetMap: using and enhancing the free map of the world*. UIT Cambridge Cambridge, 2011.

[18] J. Ziegler *et al.*, "Making bertha drive? an autonomous journey on a historic route," *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 2, pp. 8–20, 2014.