LIDAR-Based Lane Marking Detection For Vehicle Positioning in an HD Map

Farouk Ghallabi, Fawzi Nashashibi, Ghayath El-Haj-Shhade, Marie-Anne Mittet

LIDAR-Based Lane Marking Detection For Vehicle Positioning in an HD Map

Farouk Ghallabi*†, Fawzi Nashashibi*, Ghayath EL-HAJ-SHHADE† and Marie-Anne MITTET†

*INRIA Paris-Rocquencourt
Paris, France
{farouk.ghallabi, fawzi.nashashibi}@inria.fr,

†Renault s.a.s
Guyancourt, France
{ghayath.el-haj-shhade, marie-anne.n.mittet}@renault.com

Abstract—**Accurate self-vehicle localization is an important task for autonomous driving and ADAS. Current GNSS-based solutions do not provide better than 2-3 m in open-sky environments [1]. Moreover, map-based localization using HD maps became an interesting source of information for intelligent vehicles. In this paper, a Map-based localization using a multi-layer LIDAR is proposed. Our method mainly relies on road lane markings and an HD map to achieve lane-level accuracy. At first, road points are segmented by analysing the geometric structure of each returned layer points. Secondly, thanks to LIDAR reflectivity data, road marking points are projected onto a 2D image and then detected using Hough Transform. Detected lane markings are then matched to our HD map using Particle Filter (PF) framework. Experiments are conducted on a Highway-like test track using GPS/INS with RTK correction as ground truth. Our method is capable of providing a lane-level localization with a 22 cm cross-track accuracy.**

I. INTRODUCTION

Precise vehicle localization is an important task for autonomous driving and advanced driver assistance systems (ADAS). Indeed, precisely estimating the position of the ego-vehicle with respect to the driving lane is crucial for many ADAS functionalities such as Lane Keeping Assist (LKA). This localization level has been introduced in [2] as «WHEREINLANE» application. The report [2] shows that such an application requires a localization accuracy of $50cm$ which is not affordable by the majority of current localization systems. Existing localization approaches can be divided into two categories: global and relative localization [3]. Relative localization techniques are mainly focusing on pure odometry such as visual odometry [4], laser odometry [5] or SLAM-Based approaches [6], [7] such as Visual-based SLAM [8] and Laser-based SLAM [9], [10]. Global localization is mostly determined by global navigation satellite systems (GNSS). However, the position accuracy of automotive GNSS solutions is no better than 2-3 m in open-sky environments [1], which is considered to be insufficient for autonomous driving. Alternatively, localization on a prior map, usually known as Map-Based Localization or Map-matching, has gained interest since maps can be pre-built very accurately. Current Map-based localization can be divided according to the type of the sensor used: passive or active. Vision-based (passive sensor) techniques [11], [12], [13], [14], [15] match visual features or landmarks to map attributes. Visual landmarks can be of different types: lane sides [13], traffic signs painted on the road such as arrows, pedestrian crossing and speed limits [15], traffic signs (vertical poles) [16], or feature points such as SIFT [14]. Despite promising localization results, the main restrictions of vision techniques are the dependency on external light conditions and the sensitivity to shadows and illumination noise. On the other hand, LIDAR-based approaches (active sensor) have proven to be more accurate than vision techniques. Indeed, LIDAR sensors are much less sensitive to light conditions and provide a 3D representation of the environment with centimetric accuracy. However, processing the whole LIDAR points is time consuming and is not suitable for real time applications. Therefore, 3D point cloud is usually projected into 2D representations such as 2D orthographic reflectivity map [17], precise height map [18], [19], or a combination with other information such as colors, curvatures and normals [20]. Similarly to vision, recent research has been conducted on lidar-based road marking detection [21], [22], [23]. Road markings are reflective objects and can be detected using intensity laser data. Since road markings occur on the road surface, it is first necessary to perform road segmentation. Road segmentation methods often rely on detecting curbs such as road-edges. Then, marking points are extracted from the road plane according to their high reflectivity values. In [24], the road and road-edge points are extracted using the elevation information of the LIDAR, and then validated on the ground plane for robustness. In [22], an occupancy grid map of elevation difference is computed. Grid cells whose elevation difference is smaller than a threshold are clustered as ground cells. The work in [25] is an extension to [26] where the authors developed a curb detection method based on ring compression analysis. At first, LIDAR point cloud are structured in a polar grid. Then, distances between rings are computed and compared to the expected distance when rings intercept a flat plane. If the computed distance is below a threshold value then the grid cell is classified as curb.

In this paper, we propose a lane markings detection method for localization within an HD map using a Multilayer LIDAR. At first, road segmentation is performed by analysing geometry of each returned layer points. Then, high reflective road points are projected onto a 2D grid which is always centred around the vehicle position and with a cell size of 0.15m ×0.15m. A line detection using Hough Transform [27]

is implemented on the resulted image. To validate our approach within a localization system, a map-matching algorithm using particle filter and lane markings as features is implemented. Experiments were conducted on a highway-like test track at high vehicle speeds ≈ 80 kph. The laser range finder used in our experiments is a Velodyne VLP-32C. Moreover, a highly accurate GPS with RTK correction signals (ixblue: ATLANS-C) was used as ground truth for evaluation purposes. Finally, a comparative analysis between our localization system and an automotive u-blox GNSS is provided based on a cross-track metric [28].

The remainder of this paper is structured as follows: Section II describes the proposed road segmentation method. Section III illustrates the developed map-matching algorithm with a brief description of the used HD map. An experimental evaluation is presented in section IV and we conclude with perspectives and future work in section V.

II. ROAD SEGMENTATION

In order to detect lane markings, it is necessary at first to segment road points from the whole point cloud since road markings are located on the road surface. Once the road is segmented, an intensity-based thresholding is applied to extract road marking points. Our segmentation method is based on a geometric analysis of the layer scan. A layer scan is a set of captured layer points from a full physical rotation of the LIDAR. Under the assumptions of flatness and smoothness of highway road surfaces, the geometric pattern of a layer scan intercepting the road can be locally modelled by a circular arc. However, due to non smooth and discontinuous surfaces outside road boundaries such as grass, bushes and gravel, the geometry of some regions of the scan is scattered (figure 1.a) and cannot be modelled. Differently to curb-based approaches, our method is capable of detecting change of surface type (from smooth to non-smooth and vice versa) even in the absence of distinctive geometric curbs. For illustration purposes and with no loss of generality, we consider the case of one layer scan, also called ring. In addition, we consider the polar representation of points (r, θ, φ), were r is the range of the returned point, θ is the layer vertical angle and φ is the azimuth (horizontal) angle.

A. Ring geometry analysis

Let $L = \{p_i = (r_i, \theta, \varphi_i)\}$ be the set of ring points parametrized in polar coordinates. So far, L is discretized to a set of contiguous slices s_j (figure 1.b) that are defined by the following system:

$$\begin{cases} s_j = \{p_k = (r_k, \theta, \varphi_k), \varphi^j_{min} \le \varphi_k \le \varphi^j_{max}\} \\ \varphi^j_{max} = j \times \varphi_{res}, \varphi^j_{min} = (j-1) \times \varphi_{res} \end{cases} \quad (1)$$

where, φ_{res} is the slice angular resolution, $j \in \{1, 2, .., M\}$ and $2\pi = \varphi_{res} \times M$. As previously described, if a ring slice is intercepting the road then its geometry can be approximated by a circular arc. However, a slice outside the road is generally intercepting discontinuous, non flat

and non smooth surfaces such as vegetations and therefore is not geometrically consistent with a road slice model. Consequently, the proposed segmentation method is based on fitting a circular arc to each slice s_j, then checking the residual fitting score to determine whether s_j is a circular arc or not. The computation of the residual fitting score goes through three steps:

- Calculation of the mean altitude μ_z of slice points. This represents the horizontal z-plan to which the expected circular arc $\bar{s}_j$ should belong.
- Computation of the center and the radius (C_j, R_j) of $\bar{s}_j$ as follows: $C_j = (0, 0, \mu_z)$, $R_j = \mu_z \times cot(\theta)$
- Compute the normalized sum of slice point projections to $\bar{s}_j$ which is also called the fitting score value. The projection function for a given point $p_k = (r_k, \theta, \varphi_k)$ to $\bar{s}_j = (C_j, R_j)$ is:

$$proj(p_k, \bar{s}_j) = \sqrt{(R_j - r_k)^2 + (r_k \times sin(\theta) - \mu_z)^2}$$

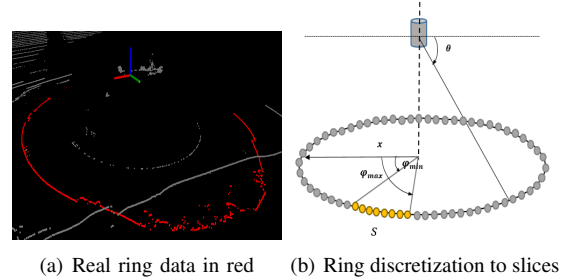(a) Real ring data in red   (b) Ring discretization to slices

Fig. 1. Slice representation of a layer scan

B. Slice-edge detection

The proposed road segmentation method makes use of the previously computed fitting score values. At first, the 2D (x, y) horizontal plan is partitioned into four quadrants $Q_{1,2,3,4}$ as depicted in figure (2.a). This partition allows us to simplify the segmentation process and to parallelize the computation. Indeed, by definition, the LIDAR ring crosses road boundaries at most once within each quadrant. Therefore, the problem can be formulated as executing a quadrant search for the slice that defines the road boundary, which we call in this paper a slice-edge.

To do that, an iterative process is implemented for each quadrant. Starting from an initial position on the ring section (blue dots in figure 2.a), the process iterates over the slices until crossing the first slice (red dots in figure 2.a) whose score value is greater than a threshold T_1. The azimuth angle that corresponds to the detected slice-edge is called: cut angle ψ. As a result, slices coming after ψ are clustered as non road slices. However, in practical situations, some slices can have high scores even when belonging to the road surface. This is due in general to a local point cloud distortion caused by vehicle dynamics or road surface irregularities. A Gaussian filter is applied to each quadrant in order to smooth the data. For example, in figure (2.b) and with a fixed threshold of $0.05m$, the cut angle (red) in the original data is approximately $42.5°$ whereas in the smoothed data (black) it is approximately $50°$.
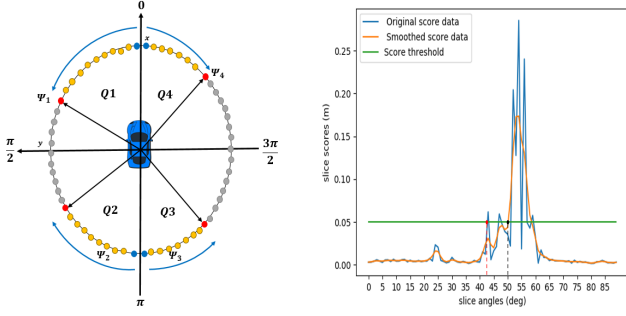
(a) Slice edges in red with their relative cut angles ψ_i, road slices in yellow

(b) Original Vs smoothed score data

Fig. 2.   Detection of slice edges

C. Road data registration

Since we are only interested in LIDAR layers that hit the road surface in the vicinity of the vehicle, we only applied the segmentation process for the first 16 layers (which are ordered according to their vertical angles: from the lowest to the highest values). Moreover, the VLP-32C Lidar, in contrary to the HDL velodyne sensors, has a low density of points on the road surface which is not sufficient for efficiently detecting road markings. To cope with this problem and taking into account high vehicle speeds, we chose to increase the spinning rate of the velodyne sensor to 20Hz (typical use is 10Hz) and we implemented a data registration process which consists in accumulating two consecutive frames of the segmented road points F_{k-1} and F_k at a given step k. With the use of inertial data, a constant velocity and yaw rate model is used to compute the transformation matrix T between F_{k-1} and F_k. The outcome of this step is a new point cloud $F_{reg} = F_{k-1} \cup T.F_k$ that has a duplicated density of road points (figure 3.b) at 10 Hz frequency.
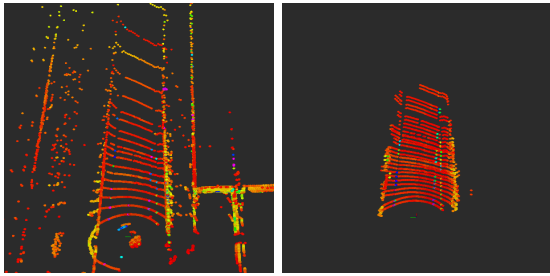
(a) Original point cloud data   (b) Result of segmentation and registration

Fig. 3.   Segmentation & registration of LIDAR data

D. Detection of lane markings

Once road points have been segmented, the intensity LIDAR data is used to extract marking points. The extraction is based on the application of an intensity threshold T_2 as we claim that most of high reflectivity returns from road surface are stemming from road markings. A 2D grid, centred around the ego-vehicle, is constructed with a cell size equals to

$0.15m$. The cell size is set taking into account road marking widths on highways. The maximum intensity value of points that fall within a grid cell is stored. Grid reflectivity data is then transformed to a 2D intensity image to which a line detection using a Hough Transform [27] is applied for lane markings detection. We chose the polar representation (r, θ) of lines as it is relevant for localization. Indeed, r and θ are interpreted as the vehicle lateral distance and heading to the lane marking. For robustness, a set of constraints are used to filter out the false detections. These constraints are summarized in the following:

1) Lines on the driving direction: the detection process only searches for lines that are approximately parallel to the driving direction of the ego-vehicle. This can be achieved by defining constraints on the θ search of the Hough Transform.

2) Parallel lines: Consider M the number of detected lines and N_k the number of parallel lines to a given line l_k. The filtering algorithm picks the line given by:

$$l_{max} = \underset{k \in \{1,..,M\}}{\arg\max} \ N_k$$

and discards all the lines that are not parallel to it.

3) Line fusion: : For two detected lane markings $l_1 = (r_1, \theta_1)$ and $l_2 = (r_2, \theta_2)$, if $\|r_1 - r_2\| < \epsilon$ then a new line is created $l_{new} = a \times l_1 + b \times l_2$ where $a + b = 1$. For our case a = b = $\frac{1}{2}$.

E. Tracking of lane markings

A tracking algorithm using Kalman filter [29] and inertial measurements has been implemented to track lane markings. A standard Kalman filter has proven to be sufficient for our case as we used a simplified prediction model given by the following:

$$\begin{cases} r_k & = r_{k-1} - v \times dt \times sin(w \times dt) + \nu_{r,k} \\ \theta_k & = \theta_{k-1} - w \times dt + \nu_{\theta,k} \end{cases} \quad (2)$$

where $\nu_{r,k} = \mathcal{N}(0, \sigma_r)$ and $\nu_{\theta,k} = \mathcal{N}(0, \sigma_\theta)$ are gaussian distributions with zero mean and variances $(\sigma_r, \sigma_\theta)$ and v, w are the vehicle speed and yaw rate, respectively. The model validity is guaranteed under the assumption of low θ values. This assumption holds for highway use cases where vehicles encounter small changes of heading while driving. Figures (4.a) and (4.b) depict the projection of the 3D road marking points to a 2D image, detected lane markings are shown in figure (4.c).

III. MAP-BASED LANE-LEVEL LOCALIZATION

The detected lane markings are used as input features to the map matching algorithm. To achieve precise vehicle localization, a High Definition (HD) Map, provided by a mapmaker, with a lane-level information and absolute accuracy of 2 cm has been used. The prototype map is composed of many links that represent lane markings. Each link is a sequence of segments expressed in WGS 84 geodetic system.
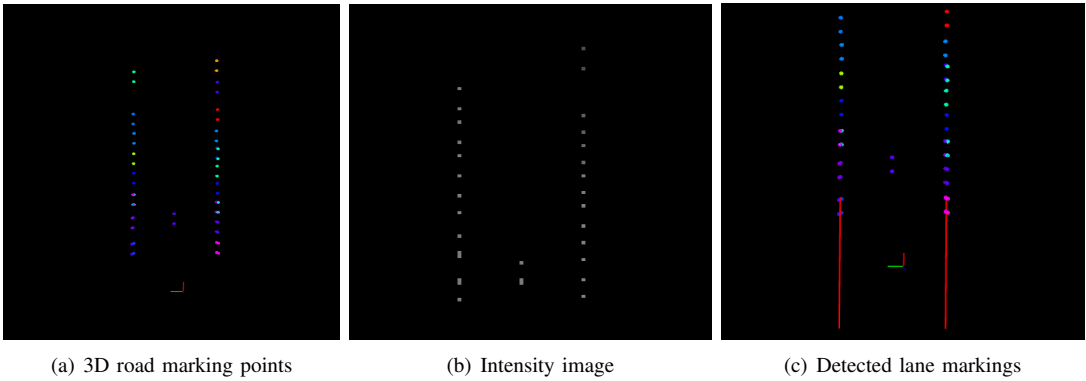
| (a) 3D road marking points | (b) Intensity image | (c) Detected lane markings |

Fig. 4.   Projection of 3D points to 2D image and lane markings detection

A. Map-matching

In order to localize the vehicle on the previously described map, a map matching algorithm has been implemented under the Particle Filter (PF) framework [30]. Particle filtering is able to represent multi-modal non-Gaussian distributions which is convenient for global localization. A set of particles (X_j, W_j) that represent vehicle pose hypotheses and their relative weights are defined. The vehicle pose is considered in our study as in 2D space: (x_j, y_j, γ_j). The particle filter algorithm is implemented in three main steps: prediction, update and resmapling steps.

For the prediction step, each particle hypothesis at time k is estimated from the previous estimation at time $k-1$ and from odometry data (velocity v and yaw rate w). Different vehicle dynamic models can be found in the literature [31]. In our case, we considered a constant velocity and yaw rate model given by the following system (3):

$$\begin{cases} \gamma_k & = \gamma_{k-1} + w \times dt + \nu_w \\ x_k & = x_{k-1} + v \times dt \times cos(\gamma_k) + \nu_x \\ y_k & = y_{k-1} + v \times dt \times sin(\gamma_k) + \nu_y \end{cases} \quad (3)$$

where $\nu_w = \mathcal{N}(0, \sigma_w)$, $\nu_x = \mathcal{N}(0, \sigma_x)$, $\nu_y = \mathcal{N}(0, \sigma_y)$ are gaussian noise distributions. Moreover, GNSS signals are used to initialize the filter.

For the update step, particle weights are recursively updated from the detected lane markings and the prototype map M. For each particle X_j, the nearest segment for each map link (lane marking) on the road section is searched. Then, its relative distance and heading with respect to X_j are computed. We will note these parameters as $m_j^j = (r_{j,l}, \theta_{j,l})$, where l is the nearest segment for a given map link. The update step is composed of two parts:

1) Weight calculation from the Map: If the projection of a particle to the map is not within the driving road section, then a zero weight is assigned.

2) Weight calculation from lane markings: for a given observation data $z_k = (r_k, \theta_k)$ and a matched link segment $m_l^j = (r_{j,l}, \theta_{j,l})$ a likelihood function similar to [13] is computed as follows:

$$p(m_l^j | z_k) = e^{\frac{(r_k - r_{j,l})^2}{2\sigma_r^2}} + e^{\frac{(\theta_k - \theta_{j,l})^2}{2\sigma_\theta^2}}$$

Where, σ_r and σ_θ define the measurement noise variances. Taking into account all the observations and all the matched links, the calculated weight is given by:

$$W^j = \alpha \sum_k \sum_l p(m_l^j | z_k)$$

Where α is a weight normalization factor.

For the resampling step a systematic resampling strategy with a threshold of 50% effective particles has been implemented to carry on particle degeneracy,. Indeed, according to [32], systematic resampling is found to provide comparable results to other state of the art methods. However, it is often preferred thanks to the simplicity of implementation.

IV. EXPERIMENTAL RESULTS

Fig. 5.   Renault test track (google earth map)

A. Experimental setup

Experiments were conducted on a double lanes Highway test track of approximately 5 km long (figure 5). Our prototype vehicle is equipped with a GPS/IMU (ixblue: ATLANS-C) localization unit with RTK correction signals of 5cm absolute accuracy, a Velodyne VLP-32C laser scanner (10-20 Hz, 32 laser beams) and an automotive GPS/IMU (u-blox B78-ADR). All sensors are synchronized to GPS time clock and data were collected for different vehicle speeds (Table II) to evaluate the robustness of our approach. The map matching algorithm has been developed under the ROS framework.

B. Road marking detection

For the road segmentation part, a histogram-based analysis was conducted to set the threshold T_1 (figure 6). The histogram illustrates the score values for Q_1-slices. One can notice that the histogram peak corresponds to slices for which the score values are less than $0.1m$, which is expected as most of the slices lie on the road surface. Consequently, the segmentation threshold value T_1 is set to $0.1m$.
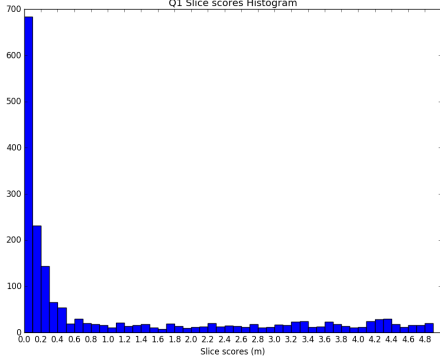

Fig. 6.   Q1-Slice variances histogram

Once 3D road points are segmented, LIDAR reflectivity byte values (from 0 to 255) are used to segment road marking points. The reflectivity threshold used in our approach is fixed with respect to the sensor datasheet which assigns a value greater than $T_2 = 100$ for retroreflector objects such as road markings.

Road marking points are projected onto a 2D grid, centered around the vehicle position, of size $[0\ m, 20\ m] \times [-15\ m, 15\ m]$ and a cell resolution of $0.15\ m \times 0.15\ m$. Hough transform is only applied for small θ values (from $-\frac{\pi}{10}$ to $\frac{\pi}{10}$) to insure detection of nearly parallel lines. The threshold distance for line fusion is set to $0.5m$.

C. Cross-track error results

In this paper, we adopted a cross-track metric presented in [28] to evaluate our localization system. To the best of our knowledge, most of state of the art localization systems are evaluated in terms of absolute error. In our approach, we propose an evaluation based on a cross-track metric since it is more relevant for localization within an HD map.

As previously stated, the map is composed of many links (lane markings) in polylines form. For each position: p_{gt} (ground truth), p_{est} (our estimation) and p_{ubx} (u-blox) , the shortest signed distances to map links of the driving road section are computed, let them be $(l_1^i, l_2^i, ..l_N^i)$, where the superscript i denotes one of the three positions. The evaluation of the cross-track error for our estimate and the u-blox position is given by the following:

$$e_p = \frac{1}{N} \sum_{i=1}^{N} (l_i^{gt} - l_i^p)$$
$$p = \{\, p_{est},\ p_{ubx} \,\}$$

Figures 7.a and 7.b depict histograms and time variations of the cross-track error for the proposed method and the u-blox system for a vehicle speed = 80 kph. Means and standard deviations are also summarized in Table I. One can notice that, with the use of the laser-based lane markings and inertial data, the mean error is reduced from -1.42 m to 0.04 m and the standard deviation from 1.0 m to 0.22 m.

TABLE I
CROSS-TRACK ERROR: MEAN AND STD

	mean (m)	std (m)
markings only	0.04	0.22
u-blox only	-1.42	1.0
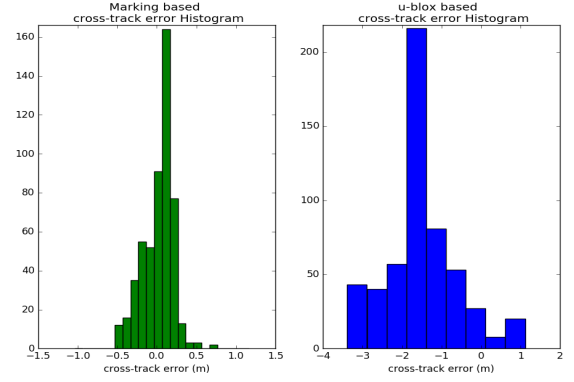
(a) Cross-track error histograms
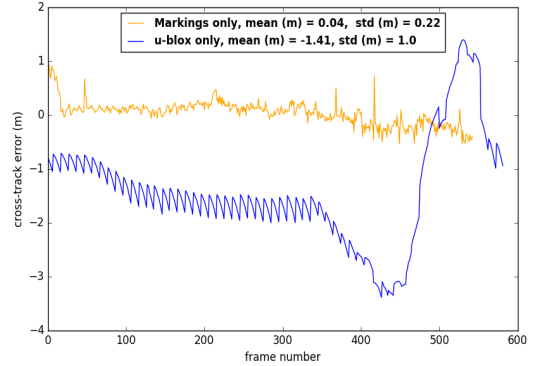
(b) Cross-track error with respect to frame number: u-blox (blue), our approach (orange)

Fig. 7.   Histograms (a) and timeplot (b) of the cross-track errors

Finally, a velocity dependent error evaluation is given in Table II. This is particularly interesting to evaluate the effect of point cloud distortion of spinning LIDARs in motion.

Results of Table II demonstrate that we have similar performances for low and high speed scenarios. This justifies that our road segmentation and lane marking detection algorithms are robust to point cloud distortion by cause of the vehicle movement. More precisely, since the main movement of the vehicle is in the x-direction (small displacement in y-

TABLE II
CROSS-TRACK ERROR VS VEHICLE SPEED

vehicle speed (kph)	mean (m)	std (m)
30	0.0	0.18
50	0.01	0.21
80	0.04	0.22

direction), lane marking positions are not affected by the point cloud distortion.

V. CONCLUSION AND FUTURE WORK

In this paper, a lane-level localization within an HD map is presented. A LIDAR sensor has been used to detect lane markings and to match them with a prototype map. The lane marking detection is implemented in two different steps: road segmentation and Hough transform on an intensity image. Road segmentation is based on Ring geometric analysis which measures the discrepancy of real local geometric features with expected features on the road surface. This is particularly useful to remove environment regions where surfaces are not smooth and are discontinuous (e.g vegetations, gravel). Furthermore, a Hough line transform using an a priori information on the environment is applied to detect lane markings. Finally, a map-matching algorithm has been implemented to validate the detection phase. Current results are promising and sufficient for Highway use-cases.

As a perspective, we believe that the road segmentation method can be advantageous to many other applications that are not covered by this study. For example, a 3D marking map can be directly built from the 3D marking points.

Future work will deal with the longitudinal localization (the along-track error) by using a monocular camera in addition to the LIDAR to detect more Highway landmarks such as traffic signs and poles. Indeed, in contrast to urban environments, highways lack reliable landmarks for longitudinal localization. Thus, we expect that minimizing the along-track error to be more challenging.

REFERENCES

[1] J.-M. Zogg, *GPS: Essentials of Satellite Navigation: Compendium: Theorie and Principles of Satellite Navigation, Overview of GPS/GNSS Systems and Applications.*, 2009

[2] EDMap, "Enhanced Digital Mapping Project Final Report," 2004

[3] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust Monte Carlo Localization for Mobile Robots," *Artif. Intell. J.*, vol. 101, pp. 99–141, 2001

[4] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," *Proc. 2004 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, pp. 652–659, 2004

[5] J. Zhang and S. Singh, "LOAM : Lidar Odometry and Mapping in Real-time," *Robot. Sci. Syst.*, 2014.

[6] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part I," *IEEE Robot. Autom. Mag.*, vol. 13, no. 2, pp. 99–108, 2006.

[7] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): Part II," *IEEE Robot. Autom. Mag.*, vol. 13, no. 3, pp. 108–117, 2006

[8] J. Fuentes-Pacheco, J. Ruiz-Ascencio, and J. M. Rendón-Mancha, "Visual simultaneous localization and mapping: a survey," *Artif. Intell. Rev.*, vol. 43, no. 1, pp. 55–81, 2012

[9] F. Moosmann and C. Stiller, "Velodyne SLAM," in *IEEE Intell. Veh. Symp. Proc.* IEEE, jun 2011, pp. 393–398

[10] A. Nuchter, K. Lingemann, and J. Hertzberg, "6D SLAM-3D Mapping Outdoor Environments," *J. F. Robot.*, vol. 24, pp. 699–722, 2007.

[11] M. Buczko and V. Willert, "Efficient Global Localization Using Vision and Digital Offline Map," pp. 1689–1694, 2017.

[12] Z. Zhu, T. Oskiper, S. Samarasekera, R. Kumar, and H. S. Sawhney, "Real-time global localization with a pre-built visual landmark database," *26th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR*, 2008

[13] F. Chausse, J. Laneurit, and R. Chapuis, "Vehicle localization on a digital map using particles filtering," *IEEE Intell. Veh. Symp. Proc.*, vol. 2005, pp. 243–248, 2005.

[14] J. Košecká, F. Li, and X. Yang, "Global localization and relative positioning based on scale-invariant keypoints," *Rob. Auton. Syst.*, vol. 52, no. 1, pp. 27–38, 2005

[15] T. Wu and A. Ranganathan, "Vehicle localization using road markings," in *IEEE Intell. Veh. Symp. Proc.* IEEE, jun 2013, pp. 1185–1190

[16] H. Li, F. Nashashibi, and G. Toulminet, "Localization for intelligent vehicle by fusing mono-camera, low-cost GPS and map data," in *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC.* IEEE, sep 2010, pp. 1657–1662

[17] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," *Robot. Autom. (ICRA), 2010 IEEE Int. Conf.*, pp. 4372–4378, 2010

[18] H. Fu, L. Ye, R. Yu, and T. Wu, "An efficient scan-to-map matching approach for autonomous driving," in *2016 IEEE Int. Conf. Mechatronics Autom. IEEE ICMA 2016.* IEEE, aug 2016, pp. 1649–1654

[19] E. Pollard, J. Perez, and F. Nashashibi, "Step and curb detection for autonomous vehicles with an algebraic derivative-based approach applied on laser rangefinder data," *IEEE Intell. Veh. Symp. Proc.*, pp. 684–689, 2013

[20] Z. J. Chong, B. Qin, T. Bandyopadhyay, M. H. Ang, E. Frazzoli, and D. Rus, "Mapping with synthetic 2D LIDAR in 3D urban environment," *IEEE Int. Conf. Intell. Robot. Syst.*, pp. 4715–4720, 2013

[21] A. Hata and D. Wolf, "Road marking detection using LIDAR reflective intensity data and its application to vehicle localization," *17th Int. IEEE Conf. Intell. Transp. Syst.*, pp. 584–589, 2014

[22] S. Kammel and B. Pitzer, "Lidar-based lane marker detection and mapping," *IEEE Intell. Veh. Symp. Proc.*, pp. 1137–1142, 2008

[23] B. He, R. Ai, Y. Yan, and X. Lang, "Lane marking detection based on convolution neural network from point clouds," in *2016 IEEE 19th Int. Conf. Intell. Transp. Syst.* IEEE, nov 2016, pp. 2475–2480

[24] W. Zhang, "LIDAR-based road and road-edge detection," in *IEEE Intell. Veh. Symp. Proc.* IEEE, jun 2010, pp. 845–848

[25] A. Y. Hata, F. S. Osorio, and D. F. Wolf, " "Robust Curb Detection and Vehicle Localization in Urban Environments"

[26] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, D. Johnston, S. Klumpp, D. Langer, A. Levandowski, J. Levinson, J. Marcil, D. Orenstein, J. Paefgen, I. Penny, A. Petrovskaya, M. Pflueger, G. Stanek, D. Stavens, A. Vogt, and S. Thrun, "Junior: The stanford entry in the urban challenge," *Springer Tracts Adv. Robot.*, vol. 56, pp. 91–123, 2009

[27] R. O. Duda and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Commun. ACM*, vol. 15, no. 1, pp. 11–15, 1972

[28] E. Héry, S. Masi, P. Xu, and P. Bonnifait, "Map-based Curvilinear Coordinates for Autonomous Vehicles," pp. 1699–1705, 2017.

[29] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *J. Basic Eng.*, vol. 82, no. Series D, pp. 35–45, 1960

[30] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte carlo localization for mobile robots," *Robot. Autom.*, 1999

[31] R. Schubert, C. Adam, M. Obst, N. Mattern, V. Leonhardt, and G. Wanielik, "Empirical evaluation of vehicular models for ego motion estimation," *IEEE Intell. Veh. Symp. Proc.*, no. Iv, pp. 534–539, 2011.

[32] O. C. R. Douc, "Comparison of resampling schemes for particle filtering," *ISPA 2005. Proc. 4th Int. Symp. Image Signal Process. Anal.*, pp. 64–69, 2005.