Feature Detection for Vehicle Localization in Urban Environments Using a Multilayer LIDAR

Alberto Y. Hata and Denis F. Wolf

***Abstract*—Localization is an important component of autonomous vehicles, as it enables the accomplishment of tasks, such as path planning and navigation. Although vehicle position can be obtained by GNSS devices, they are susceptible to errors and satellite signal unavailability in urban scenarios. Several map-aided localization solution methods have been proposed in the literature, but mostly for indoor environments. Maps used for localization store relevant environmental features that are extracted by a detection method. However, many feature detection methods do not consider the presence of dynamic obstacles or occlusions in the environment, which can impair the localization performance. In order to detect curbs even in occluding scenes, we developed a method based on ring compression analysis and least trimmed squares. For road marking detection, we developed a modified version of the Otsu thresholding method to segment road painting from road surfaces. Finally, the feature detection methods were integrated with a Monte Carlo localization method to estimate the vehicle position. Experimental tests in urban streets have been used to validate the proposed approach with favorable results.**

***Index Terms*—Vehicle localization, curb detection, road marking detection and LIDAR sensor.**

I. Introduction

AUTONOMOUS cars are comprised of fundamental systems, such as obstacle detection, navigation and localization, which enable a vehicle to travel safely in streets. Localization plays an important role, as it determines the vehicle position in the environment. Based on localization information, we can compute the position of an obstacle and perform the path planning. Therefore, the correctness and accuracy of the localization system impact the functionality of an autonomous vehicles.

An accurate position can generally be obtained by commercially available Global Navigation Satellite System (GNSS) devices. However, obstacles, as buildings, trees and tunnels may block the satellite signal reception and make GNSS impractical in urban places. A way to overcome this problem is using a map-aided localization. In this approach, sensor data are matched with the map and the positions with higher correspondence represent probable vehicle positions. Environmental features like roads, buildings and signalizations are commonly used to represent the map; therefore their correct detection are also important for a precise vehicle localization.

Urban features are frequently used for map representation and localization. In [1] building facades were used as localization features and associated with GIS maps for the localization of a vehicle. On the other hand, in [2] curb structures were used as map features and applied for localization. Basically, curbs were detected by the analysis of the height variation of a single 2D LIDAR scan. Later, the obtained curb points are registered into an occupancy grid map and the obtained map integrated to a Monte Carlo Localization (MCL) method. Differently, Gruyer *et al.* [3] adopted lane markings as map features. Intensity data of left and right camera images were analyzed and a polynomial fitting was applied for the extraction of lanes. The generated map was then combined with the Extended Kalman Filter (EKF) localization algorithm. Some authors combine two or more features for higher precision in the localization. The work of [4] combines road marking and building structure data in a map for localization. Other works have combined curbs and road marking. In [5], multilayer LIDAR data and camera images were used to detect road marking and curbs and EKF was employed for localization. Deusch *et al.* [6] used a similar approach, but the Maximally Stable Extremal Region method was used for the extraction of features from curb and road markings. Localization was performed through the Particle Filter method.

All such methods rely on camera images to detect road markings. Although cameras can return color information, they are sensible to light variation or lack of illumination. On the other hand, LIDAR sensors can work independently of light conditions and directly return distance information. Besides, few curb detection methods, as Siegemund, *et al.* [7] and Oniga [8], have handled the presence of dynamic obstacles or occluding scenes. In spite of using temporal information as these methods, we just need to analyze a single frame to remove the outliers. In regards to existing road marking detection methods, they work only for lane markings [9] and rely on a fixed threshold to separate the asphalt intensity from road marking intensity [10]. We propose a method that can extract any type of marking and avoid the constant adjustment of the threshold.

This paper presents a vehicle localization method robust to occlusions and invariant to illumination, thanks to the proposed novel feature detectors. A single multilayer LIDAR is used for the environment perception. The environment map used for localization includes curbs and road markings.

The main contribution of this paper is the curb detector that makes use of robust regression to deal with occlusions and the

Manuscript received November 30, 2014; revised May 5, 2015 and July 4, 2015; accepted September 4, 2015. Date of publication September 28, 2015; date of current version January 29, 2016. This work was supported in part by FAPESP under Grants 2012/02354-1 and 2014/09096-3 and in part by the Mobile Robots Laboratory Team. The Associate Editor for this paper was M. Bertozzi.

The authors are with the Mobile Robotics Laboratory, University of São Paulo, 13560-970 São Carlos-SP, Brazil (e-mail: hata@icmc.usp.br; denis@icmc.usp.br).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2015.2477817

road marking detector that is based on a modified Otsu method to analyze LIDAR reflective intensity data. These features are stored in a high-resolution occupancy grid map and an MCL that handles multilayer LIDAR data is used for localization. Experiments have been conducted in urban streets to validate the methods.

The paper is structured as follows: Section II describes the curb and road marking detection methods; Section III addresses the occupancy map building process and the MCL method based on road features; Section IV provides the experimental results; Section V draws the conclusions.

II. Road Feature Detection

Relevant features must be stored in a map for a map-aided localization. In this study curb and road markings were chosen as features for the construction of maps, because they are usually present in urban cities.

A multilayer LIDAR has been employed for the extraction of road features. This sensor can emit LASER beams from different angles and produces 3D information on the environment in a single scan.

Here we have adopted the Velodyne HDL-32E LIDAR model, which emits 32 beams. In a single reading, it returns a point cloud formed by 32 concentric measurements, named here "rings." However, as the outer rings become more sparse, we restricted to use 21 internal beams. For this configuration, the maximum detection distance is approximately 33 m.

The next sections describe the proposed curb and road marking detection methods.

A. Curb Detection

An obstacle can be detected by multilayer LIDAR in several ways. Montemerlo *et al.* [11] analyzed the distance between consecutive rings formed by the sensor reading, therefore, distances lower than a threshold were classified as obstacles. In an urban environment, obstacles can be roughly grouped according to the ring compression (e.g. walls have higher ring compression in comparison to a road). The proposed curb detector extends the obstacle detector of [11] and introduces filters to remove false positives. These filters check if the detected obstacle has curb-like features and handle occlusions. Details of the curb detector are provided in the following subsections.

1) Ring Compression Analysis: In order to determine the ring compression range that represents curb-like obstacles, first we must compute the regular distance between rings (i.e. in a scenario where LIDAR beams intercept a flat plane). In Fig. 1, the ring radius is given by

$$r_i^{\text{plane}} = \frac{h}{\tan\theta_i} = h\cot\theta_i \tag{1}$$

where r_i corresponds to the i-th ring, h is the sensor height relative to the plane and θ_i is the laser beam angle. Therefore, the distance between consecutive rings i and $i+1$ is calculated by the following expression

$$\Delta r_i^{\text{plane}} = r_{i+1}^{\text{plane}} - r_i^{\text{plane}} \tag{2}$$

$$= h(\cot\theta_{i+1} - \cot\theta_i) \tag{3}$$

where $\Delta r_i^{\text{plane}}$ denotes the compression of ring i on a plane.

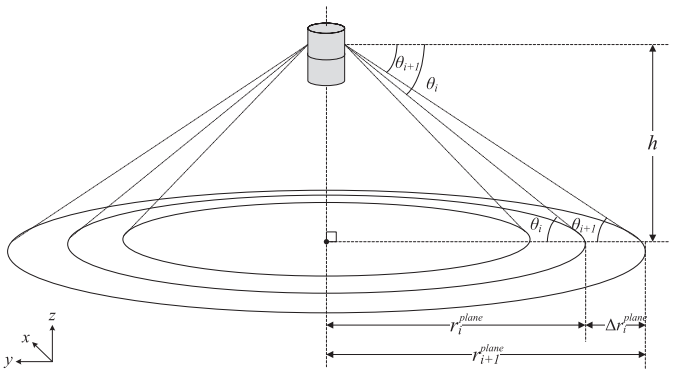


Fig. 1. Multilayer LIDAR rings intercepting a flat plane.

When a laser beam intercepts an obstacle, the distance between rings Δr_i becomes smaller than $\Delta r_i^{\text{plane}}$ and curbs are detected by checking if Δr_i is in a interval based on $\Delta r_i^{\text{plane}}$

$$S_i = \left[\alpha\Delta r_i^{\text{plane}}, \beta\Delta r_i^{\text{plane}}\right) \tag{4}$$

where S_i is the interval used to check if a point belonging to the i-th ring is a curb. Parameters α and β adjust the range bounds, with $\beta > \alpha$.

After range S_i has been determined, the distance between adjacent LIDAR points belonging to consecutive rings must be compared with S_i. To simplify the comparison, the LIDAR data are placed into a virtual circular grid with i rows and j columns. Each row i stores LIDAR points of a single ring. Each cell $c_{i,j}$ of this grid stores the mean value of the position of points $(x_{i,j}, y_{i,j}, z_{i,j})$ that fall in $c_{i,j}$ and the mean Euclidean distance $d_{i,j}$ from the mean position to the sensor (x, y and z correspond respectively to the vehicle longitudinal, lateral and upward direction).

Using the circular grid, distances between adjacent ring points are calculated by

$$\Delta d_{i,j} = |d_{i+1,j} - d_{i,j}| \tag{5}$$

where $\Delta d_{i,j}$ is the compression of the j-th cell of the i-th ring and $d_{i,j}$ represents the distance to the sensor stored in the same cell. Cells with $\Delta d_{i,j}$ value within interval S_i are classified as curb candidates.

2) Gradient Filter: The analysis of the distance between rings may cause a misclassification when we face obstacles of similar slope as curbs.

The differential filter checks the height variance in the lateral (y) direction. As curbs show a relative large height variance in the y direction and low height variance in the x direction, the gradient filter considers these two components. Gradient $\nabla g_{i,j}$ of cell $c_{i,j}$ is computed by the following equation:

$$\nabla g_{i,j} = \frac{z_{i,j+1} - z_{i,j-1}}{\text{dist}(c_{i,j+1}, c_{i,j-1})} \tag{6}$$

where $z_{i,j}$ is the height value of a cell $c_{i,j}$ and $\text{dist}(a,b)$ is the Euclidean distance function between points a and b in the (x,y) Cartesian coordinate reference.

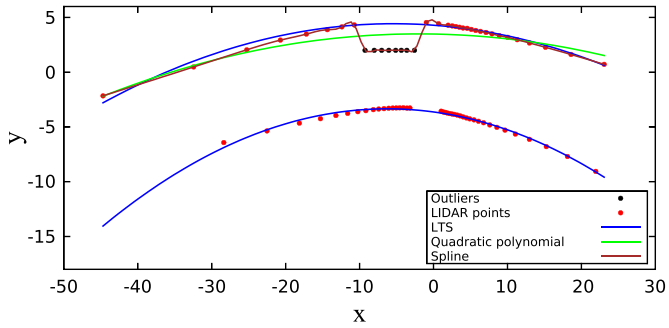


Fig. 2. Comparison between LTS and other curve fitting methods for the estimation of the curb shape.

TABLE I
COMPARISON OF THE PERFORMANCE OF DIFFERENT FITTING METHODS FOR THE ESTIMATION OF THE CURB MODEL

	SSE	R^2	Dist. to Outllier
Least squares (quad. pol.)	27.68531	0.59519	1.33402
Spline ($p = 0.99$)	0.48798	0.99286	0.07516
LTS (quad. pol.)	1.33002	0.97789	2.23326

For each cell $c_{i,j}$, we compute the $\nabla g_{i,j}$ and compare with a t_s threshold to classify it as a curb candidate or not.

3) Occlusion Removal: The gradient filter can incorrectly detect pedestrians and cars present in the streets as curb candidates. Furthermore, these obstacles cause occlusion to the LIDAR sensor and interfere with the curbs detection. The regression filter was introduced to make the detection robust to occlusions.

Basically, the regression filter fits a function over the curb candidate points and removes the outlier points. We applied the least trimmed squares (LTS) robust regression as the fitting method for the regression filter. In comparison to other fitting methods, LTS computes the function parameters considering the presence of outliers. Fig. 2 shows the curve of road data estimated by three different fitting methods and Table I provides their performance. Although the spline fitting obtained the best sum of squared errors of prediction (SSE) and the coefficient of determination (R^2), it showed the shortest distance to the outliers. The performance of the LTS regression was slightly lower than that of the spline, however, satisfactory curve models were obtained in the presence of outliers. Compared to another outlier removal method such as RANSAC, LTS obtain similar results; however, the later one is optimized to work with polynomials.

The LTS method is based on the least squares regression theory. One of its advantage is that it computes the function parameters of the data subset that minimizes the cumulative error (i.e. the subset that contains no outliers) [12]. In the LTS fitting, a dataset with n points and h is the subset that does not contain outliers $((n/2) < h < n)$ and the function parameters are obtained by the minimization of the following function:

$$f_{\text{lts}}(\gamma) = \arg\min_{\gamma} \sum_{i=1}^{h} r_i^2(\gamma) \tag{7}$$

where r is the residual function and γ is the curve function parameters. Here we adopted the quadratic polynomial for the LTS fitting.

The curve parameters are computed independently for left and right road sides. After the curve parameters have been obtained, the condition in Eq. (8) is used to preserve only curb candidates whose distance is shorter than a given threshold t_d.

$$f_{\text{lts}}(x_{i,j}) < t_d \tag{8}$$

where $x_{i,j}$ is the longitudinal component of cell $c_{i,j}$.

4) Parameter Tuning: We can exhaustively test several values for parameter set $\theta = \{\alpha, \beta, t_s, t_d\}$ to obtain satisfactory results in the curb detection. As this approach is impractical and the best parameters may not be achieved, we automatized the parameter search through a tool named Meta-Evolver,[1] in which an objective function must be defined for the establishment of fitness of parameters. The following equation was adopted:

$$f_{\text{tun}}(\theta) = \arg\max_{\theta} \frac{n}{\text{MSE}}(\theta) \tag{9}$$

where MSE is the mean squared error of the estimated curb model in relation to the ground truth (actual curb model) and n corresponds to the number of points detected as curbs (after the filtering process). The MSE is defined as

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} \left(\text{pos}_i^{\text{detec}} - \text{pos}_i^{\text{real}}\right)^2 \tag{10}$$

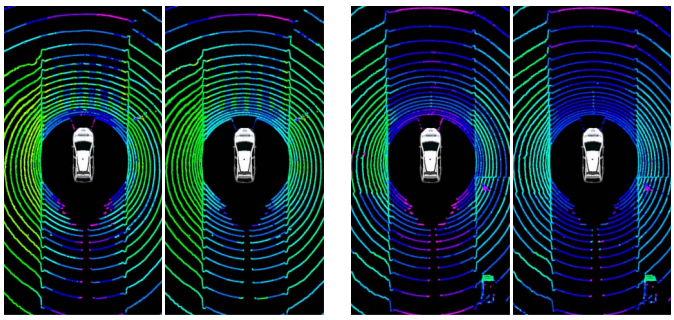
where $\text{pos}^{\text{detec}}$ refers to the curb point detected by the current parameter values and pos^{real} is the actual curb point position which is obtained by manually classifying the LIDAR data. A situation is considered optimal when many points have been detected as curbs and a low MSE has been obtained.

The first step of Meta-Evolver is to take a random walk in the parameter space based on arbitrary values of θ. After some values of θ and the corresponding f_{tun} have been obtained, a linear function is fitted in these values. The direction in which θ values must be explored is given by the resulting linear function. After a number of iterations has been reached or if no changes have occurred in the values of θ, the search is stopped and the resulting parameters are used in the curb detector method.

B. Road Marking Detection

The LIDAR reflective intensity data correspond to the amount of power necessary for the sensor to receive back an emitted beam. With this information, rough and shiny surfaces can be distinguished. The same idea can be used to segment road marking points from asphalt points. Here, the Otsu thresholding method, which is an image processing method that segments background from foreground in gray-scale images, was adapted to work with intensity point cloud data. The following sections describe the sensor intensity calibration and the proposed road marking detection method.

[1] http://sourceforge.net/projects/annevolve



(a) (b)

Fig. 3. Result of deterministic intensity calibration of two marking types. For (a) and (b) images, left is the original scenario and right the calibrated intensity. The asphalt and the road paintings intensity values become more uniform after the calibration. The asphalt is related to blue color and the road marking to green color. (a) Crosswalk. (b) Dashed line.

1) LIDAR Reflective Intensity Calibration: Due to the sensor manufacturer's miscalibration [13], the returned intensity may not correspond to the expected values. To exemplify it, all LIDAR beams that have intercepted a uniform asphalt surface must return the same intensity value, which does not usually occur. Therefore, the calibration of the intensity return is required for the detection of road markings.

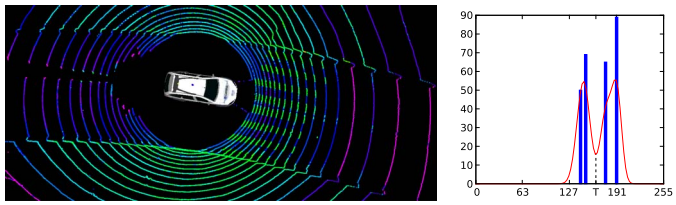
Among the existing calibration methods, we have adopted the deterministic calibration proposed by [13], as it is non-parametric and incremental and depends only on intensity log data collected in an arbitrary place. It is comprised of the following steps:

1) Obtaining LIDAR points by traversing an arbitrary environment and their association with position data.
2) Placing the LIDAR points onto a 2D grid. Each grid cell stores the list of points that fall in the same cell. Each point is represented by a pair (j, a), where a is the intensity value returned by a beam with index j.
3) Seeking for all cells in the grid that contain pair (j, a). For all cells found, the average intensity is computed over all point lists, but points whose beam is j are ignored. The calibrated intensity of (j, a) is the resulting average.

The calibration is performed for all pair combinations of j and a values. In the absence of an intensity value in the grid, the calibrated return is computed by interpolating values with close intensities. Here, the calibration resulted in a 32×256 table that corresponds to the number of LIDAR beams and the intensity range. Later, for every LIDAR point received, the intensity is corrected by the verification of the correct response in the calibration table. Fig. 3 shows the LIDAR intensity data before and after the calibration in two different road markings. The Otsu road marking detection is then applied to the calibrated points, as described in the next section.

2) Otsu Road Marking Detection: We use the left $f_l(x)$ and right $f_r(x)$ curve models obtained by the curb detector (calculated by regression filter) to restrict the analysis to the road area. The road surface is retrieved by the extraction of points inside the $f_l(x) < y < f_r(x)$ interval.

The analysis of the calibrated LIDAR data of a road surface (Fig. 4(a)) shows the intensity values of points belonging to



(a) Velodyne calibrated intensity (b) Intensity histogram

Fig. 4. Intensity histogram of a ring that has intercepted a road marking. (a) The intensity values of the road marking and the asphalt are visually distinct (green and blue). (b) A LIDAR ring that intercepted the road produces a histogram with two modes, each one representing asphalt and road marking.

asphalt ($\approx$ 190) and road marking ($\approx$ 127) are considerably distinct. The intensity histogram of such points has a bimodal shape, with one mode grouping asphalt intensities and another grouping road marking intensities (Fig. 4(b)). The Otsu thresholding method [14] is applied for the determination of the intensity value T that maximizes the variance between background (asphalt) and foreground (road marking) classes. The value T obtained is used as a threshold to segment the point cloud into asphalt and road marking.

The steps of the road marking detector are described as follows:

1) Compute the normalized histogram of the intensity: Let n_i be the i-th histogram value and M the number of road points. The normalized intensity (p_i) is computed by

$$p_i = \frac{n_i}{M}.$$

2) Compute the cumulative sum: The cumulative sum $P(k)$ is the probability of a LIDAR ring point belonging to the $[0, k]$ range and calculated by

$$P(k) = \sum_{i=0}^{k} p_i$$

where $0 < P(k) < 255$ and L is the number of possible intensity values represented by the LIDAR.

3) Compute the cumulative mean: The cumulative mean $m(k)$ represents the mean intensity of the $[0, k]$ range computed by

$$m(k) = \sum_{i=0}^{k} i p_i.$$

4) Compute the global cumulative mean: The global cumulative mean m_G is the mean intensity of the whole histogram:

$$m_G = \sum_{i=0}^{L-1} i p_i.$$

5) Compute the global variance: The global variance σ_G^2 is the intensity variance sum over road points:

$$\sigma_G^2 = \sum_{i=0}^{L-1} (i - m_G)^2 p_i.$$

6) Compute the local variance: The local variance σ_L^2 is the variance of a specific intensity:

$$\sigma_L^2 = \frac{m_G P(k) - m(k)}{P(k)\left(1 - P(k)\right)}.$$

7) Obtain the threshold: Threshold T is the value of k that maximizes σ_L^2:

$$T = \underset{0 \le k \le R-1}{\arg\max}\, \sigma_L^2(k).$$

8) Check the separability measure: The separability measure $\eta(T)$ represents the performance of T to separate the histogram into two classes. A better separability is associated with $\eta(T)$ around 1. Therefore we can verify if $\eta(T)$ is higher than a given threshold t_η.

$$\eta(T) = \frac{\sigma_L^2(T)}{\sigma_G^2}.$$

9) Check the separability intensity T: As the road marking intensity is associated with low values, T must be higher than a certain intensity t_T.
10) Check the cumulative sum of $P(T)$: Only a certain part of the roads is generally covered by painting, therefore, $P(T)$ must never be higher than a predefined threshold t_P.
11) Detect lane marking: If the point satisfies 8, 9, and 10 and the intensity point is lower than T, classify the point as road marking.

The value T can be calculated once or recalculated for each new LIDAR measurement. Due to the variance in the road marking and asphalt intensity along the streets (caused by natural wear), we recompute T for every LIDAR measurement.

For improved robustness, the sequence of points detected as road marking and longer than a certain length is discarded, as proposed by [9]. This filter is explained by the fact that road marking stripes are not wide (regarding crosswalks and lane markings).

III. Vehicle Localization

For localization, first, the detected curbs and road markings points are stored in an occupancy grid map. The generated map is then integrated with MCL method for the estimation of the vehicle position. The next sections describe the mapping process and the employed MCL method.

A. Occupancy Grid Mapping

Before the construction of the environment map, the curb and road marking points are inserted in a binary grid map of 0.10 m resolution. The position of the points is obtained by a GNSS device and using Real Time Kinematic (RTK) solution. In order to generate the occupancy grid map, a planar virtual laser scan of 1 degree resolution sweeps the same path used for the generation of the binary map. For each cell m_i, the occupancy probability at time t is calculated by

$$p(m_i|z_t, x_t) = 1 - \frac{1}{\exp(l_i)} \tag{11}$$

where z_t is the sensor reading, x_t is the vehicle pose and l_i is the cell occupancy in the log odds form

$$l_i^T = (T-1)\log\frac{1 - p(m_i)}{p(m_i)} + \sum_{T}^{t=1} \frac{p(m_i|z_t, x_t)}{1 - p(m_i|z_t, x_t)}. \tag{12}$$

The resulting occupancy grid is then applied for the vehicle localization.

B. Monte Carlo Localization

The MCL method is based on the Bayesian filter that estimates the position by matching the sensor measurements with the environment map. The vehicle position is given by a set of particles continuously updated after each sensor measurement.

The MCL algorithm starts by randomly distributing particles over the environment. The particle set is represented by $\mathbf{S}_k = \{s_k^i; i = 1, 2, \ldots, n\}$, where n is the number of particles and k represents the time stamp. Each particle s_k^i stores the position x_k^i (latitude, longitude, and orientation) and the importance weight w_k^i that represents the vehicle position certainty.

For each MCL iteration, $\mathbf{S}_k$ is updated through an auxiliary particle set $\mathbf{S}'_k$. Four steps comprise the particle set update:

1) The motion update step estimates position x_k^i of particles $s_k'^i$. The position is calculated by probability $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_{k-1})$, where $\mathbf{x}$ and $\mathbf{u}$ correspond to the position and motion of the vehicle. The motion can be obtained by dead reckoning, (IMU) Inertial Measurement Unit, (INS) Inertial Navigation System or a low-cost GPS sensor.
2) The measurement update step computes weight w_k^i of particles $s_k'^i$ from the probability of generating the road feature $\mathbf{z}_k$ given the current position x_k^i and the occupancy grid map m, trough the likelihood field model:

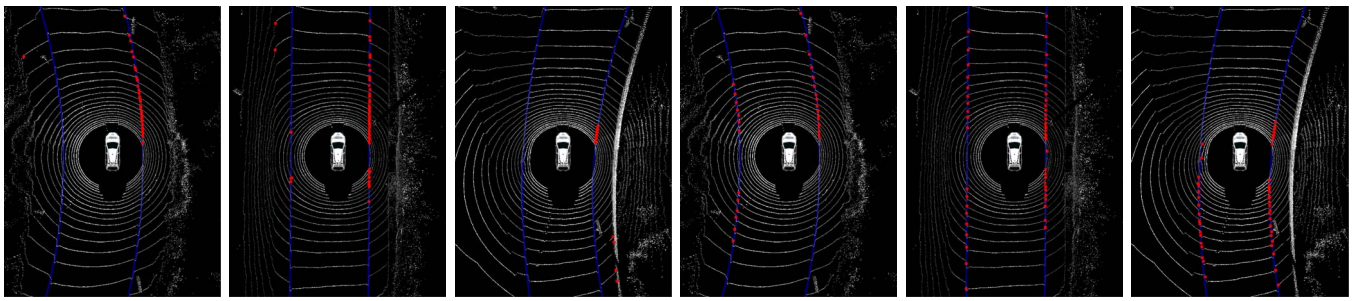
$$w_k^i = p\left(\mathbf{z}_k|x_k^i, m\right).$$

3) n particles are chosen randomly (with reposition) from $\mathbf{S}'_k$. Particles of heavier weights have proportional probability of being selected. The chosen particles replace the current $\mathbf{S}_k$ set.
4) The particle of heaviest weight w_k^i in $\mathbf{S}_k$ is chosen to represent the position of the vehicle.

The localization also works even with the absence of either curb or road marking information in certain parts of the map, because the weights are calculated based on the correspondence of the sensor with the map.

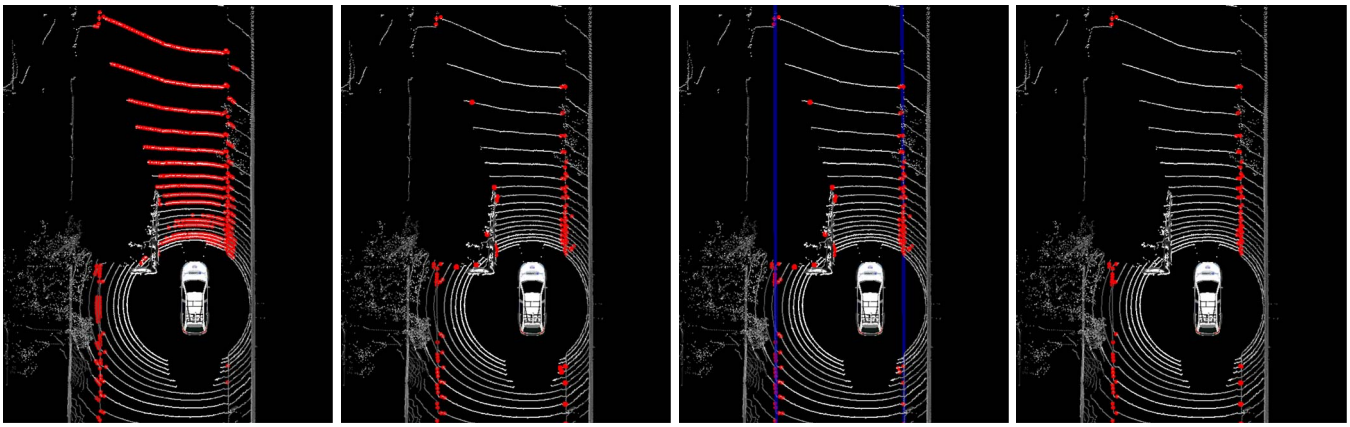
IV. Experiments and Results

Road marking detection experiments have been performed with CaRINA 2 autonomous car prototype mounted with a Velodyne HDL-32E LIDAR sensor. The sensor is mounted 2.30 m above the ground to minimize the occlusion and maximize the sensor field of view. Vehicle odometry has been obtained by Xsens MTi-G INS (Inertial Navigation System) device and the positioning was obtained by Septentrio AsteRx2i HDC GNSS device, which supports RTK solution. The GNSS



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

Fig. 5. Curb detection before (a)–(c) and after (d), (e) parameter tuning of three scenarios: left turn (a), (d), straight road (b), (e) and right turn (c), (f). Red points denote the detected curb and the blue line denote the ground truth.



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

Fig. 6. Curb detection steps in an occluded scene in a steep road. Red points denote the detected curb and the blue line denote the LTS regression. (a) Ring compression. (b) Gradient filter. (c) Regression filter. (d) Final result.

device was used only for environment map building, although less precise sensors along SLAM algorithms could be used to achieve similar results in this task. The following sections describe the experiments. The odometry information was used to compute the motion model during the MCL localization, but alternatively other sensors like wheel encoders could be used.

A. Curb Detection

Three scenarios have been considered for the estimation of the best θ values by parameter tuning: a straight road, a left turn and a right turn. For each scenario, curb points were manually marked in a previous step. θ values were modified according to Equation (9). The parameter tuning method was limited to 500 iterations and 20 executions.

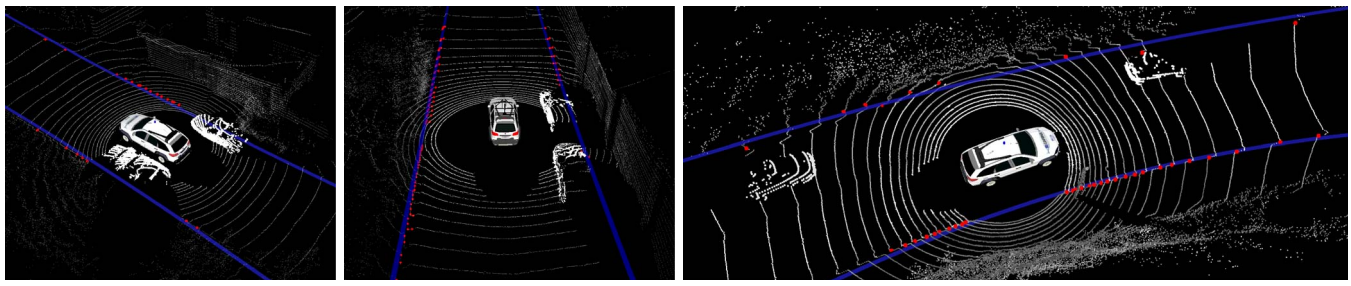
The best execution ($f_{\text{tun}} = 23033$) obtained the following values for θ: $\alpha = 0.113$, $\beta = 1.375$, $t_s = 0.124$ m and $t_d = 0.596$ m. Fig. 5 shows results of the curb detection with the parameters obtained. The curb detection MSEs of each scenario were 2.507×10^{-4} (left turn), 4.53×10^{-3} (straight road) and 1.532×10^{-3} (right turn). The parameters do not need to be retrained unless the test environment is considerably different from typical urban environments.

Fig. 6 shows the curb detection steps in a scenario with an occluding car crossing a steep street. The sensor creates a non-perpendicular angle relative to the surface because the vehicle center of mass is not in the center of the vehicle. The curb detector used the θ values obtained in the previous step. In Fig. 6(a) the ring compression analysis is prone to false positives in a non-flat terrain. The gradient filter eliminates points with slope that do not correspond to curb structures (Fig. 6(b)) and regression filter eliminates points originated by occlusion through LTS fitting (Fig. 6(c)). The curb detections in other occluding scenarios are also shown in Fig. 7.

B. Road Marking Detection

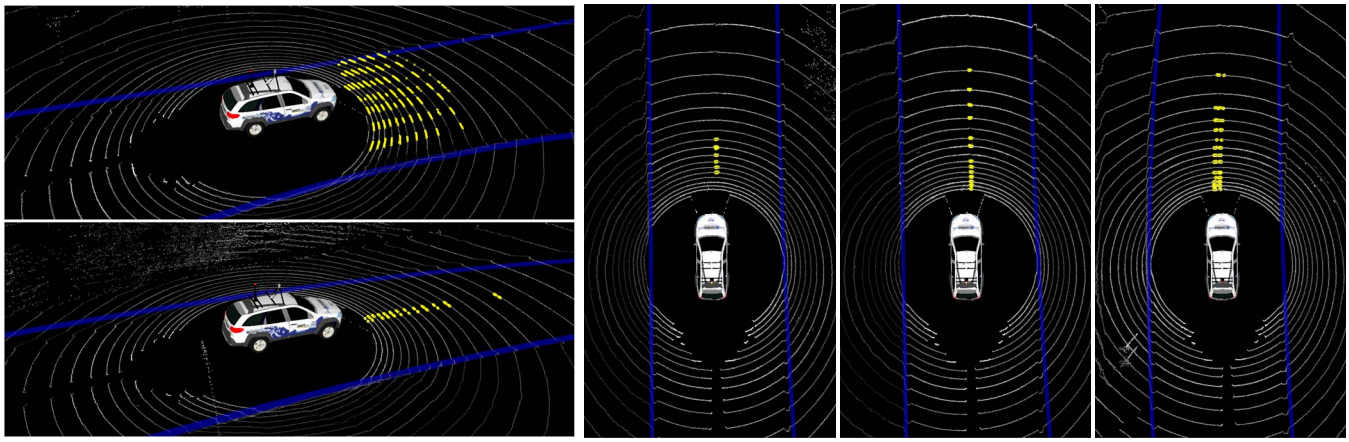
Threshold values were configured with $t_n = 0.90$ (maximum separability value), $t_T = 197$ (maximum intensity for road markings) and $t_P = 0.80$ (maximum cumulative sum) for road marking detection.

Fig. 8 shows the road marking detection in different urban streets. Blue lines represent the road boundaries obtained by the curb detector and delimit the search space of the road markings. Yellow points represent the road marking detected by the adapted Otsu method.



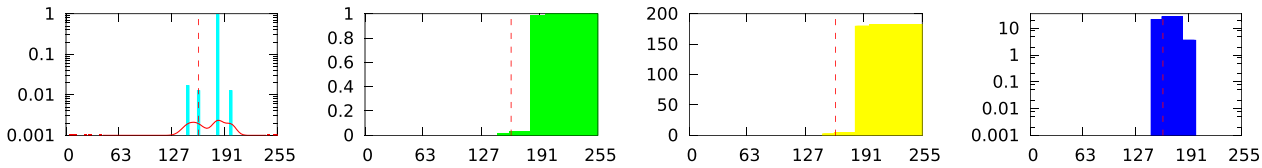
(a) (b) (c)

Fig. 7. Curb detection in different situations. (a) Cars on both street sides. (b) Two close cars on the road side. (c) Two distant cars on the road side. Red points denote the detected curb and the blue line denote the LTS regression.



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

Fig. 8. Road marking detected by a modified version of Otsu method. Blue lines represent the road bounds estimated by the curb detector and yellow points correspond to the detected road marking. (a) Top: crosswalk; Bottom: double line. (b) Dashed line. (c) Continuous line. (d) Double line.



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

Fig. 9. Intensity histogram of the LIDAR points shown in Fig. 8(b) that are inside the curb bounds. (a) Intensity histogram. (b) Cumulative sum. (c) Cumulative mean. (d) Intensity variance.

The intensity histogram aspect of the road represented in Fig. 8(b) is shown in Fig. 9. The Otsu method divided the bimodal histogram at $T = 160$, which resulted in a maximum variance of $\sigma_L^2 = 27.13$. At this point, the cumulative sum and mean show that the number of points starts to increase at value T, i.e., the starting intensity value of the asphalt surface. The values of cumulative sum and separability measure were, respectively, $P(T) = 0.0128$ and $\eta(T) = 0.8589$, which satisfied t_P and t_n.

C. Mapping of the Environment

Curb and road marking points obtained from each detection were associated with a global coordinate reference (using the GNSS device) and concatenated with each other for the mapping of the environment. The occupancy grid algorithm was then applied to the point cloud of a single track. Only the planar information was considered for the map generation and the grid resolution was configured to 0, 10 m.

Two tracks have been adopted for the map building, named here as "Track 1" and "Track 2." Curb and road marking data were obtained by traveling (clockwise direction) the blue line path represented in Fig. 10(a) and (b). The resulting occupancy grid map is illustrated in Fig. 10(c) and (d), where black, white and gray cells represent, respectively, obstacles, free space and unknown space. Details of the map are also provided in Table II.

D. Vehicle Localization

The MCL method has been employed so that the curb and road marking features could be matched against the environment map. An INSS device (Xsens MTi-G) was used for

Fig. 10. Environment maps used in the localization experiments. Track 1 (770 m length) is a 199.9 m × 306,9 m map and Track 2 (1749 m length) is a 573.0 m × 536.7 m map. The occupancy grid maps have 0.10 m resolution. (a) Track 1—Aerial map. (b) Track 1—Occ. grid map. (c) Track 2—Aerial map. (d) Track 2—Occ. grid map.

TABLE II
DETAILS OF THE TRACKS USED IN THE MAPPING AND LOCALIZATION EXPERIMENTS

	Length	Map Size	Resolution
Track 1	770 m	199.9 m × 306.9 m	0.10 m
Track 2	1749 m	573.0 m × 536.7 m	0.10 m

obtaining of the vehicle initial position and trajectory. Later, the information provided by this sensor was used to get the vehicle odometry and then generate the motion model. The localization will not be seriously affected in case of momentary GPS signal loss, because we just need an estimation of the vehicle odometry.

The MCL method was configured to use at least 100 and at the maximum 2500 particles, therefore the particle number was increased or decreased according to the localization covariance. These particles were initially distributed around the position given by the GPS device. Due to the GPS error, the vehicle was run few meters for the particle convergence.

We conducted localization experiments in scenarios of Tracks 1 and 2 and with other cars crossing the street. The vehicle traveled at 30 km/h to 45 km/h speed. The results were compared with a ground truth obtained by an accurate positioning solution (Septentrio AsteRx2i has millimeter accuracy using RTK solution).

We traversed Track 1 three times, which resulted in longitudinal, lateral and angular errors of 0.1395 m, 0.2040 m and 0.019 rad, respectively. Regarding Track 2, we traversed it twice and obtained longitudinal, lateral and angular errors of 0.2721 m, 0.2222 m, and 0.0158 m. The variances of the longitudinal, lateral and angular components were similar in both tracks. Tables III and IV show the results for Tracks 1 and 2 in each lap. The last row contains the overall result.

Fig. 11 depicts the localization results of the first lap. Fig. 11(a) and (b) represents the difference between the MCL position and the ground truth by a circle. Fig. 11(c) and (d) represents the MCL covariance in x and y directions through ellipses. Fig. 11(e) and (f) illustrates the MCL lateral, longitudinal and angular errors obtained by the comparison with the ground truth. Errors are larger in localization starting point and when the features could not be detected properly due to the presence of obstacles. Fig. 11(g) and (h) shows the particle number change along the traveled distance.

TABLE III
LATERAL, LONGITUDINAL AND ANGULAR ABSOLUTE AVERAGE ERRORS IN TRACKS 1 AND 2

(a)

Lap	x (m)	y (m)	θ (rad)
1	0.1081 ± 0.0816	0.2174 ± 0.2188	0.0204 ± 0.2681
2	0.1486 ± 0.1216	0.1756 ± 0.1350	0.0099 ± 0.0076
3	0.1647 ± 0.1171	0.2177 ± 0.1973	0.0355 ± 0.3411
1∼3	0.1395 ± 0.1102	0.2040 ± 0.1891	0.0219 ± 0.2513

(b)

Lap	x (m)	y (m)	θ (rad)
1	0.2438 ± 0.3490	0.1295 ± 0.0876	0.0113 ± 0.0111
2	0.3024 ± 0.1715	0.3217 ± 0.1922	0.0206 ± 0.0129
1∼2	0.2721 ± 0.2795	0.2222 ± 0.1761	0.0158 ± 0.0129

TABLE IV
ABSOLUTE AVERAGE MCL VARIANCE AND PARTICLE NUMBERS IN TRACKS 1 AND 2

(a)

Lap	$\Sigma(x)$ (m^2)	$\Sigma(y)$ (m^2)	$\Sigma(\theta)$ (rad^2)	Particles
1	0.01345	0.03308	0.00083	182
2	0.01050	0.00823	0.00074	181
3	0.01145	0.00695	0.00091	245
1∼3	0.01161	0.01479	0.00081	205

(b)

Lap	$\Sigma(x)$ (m^2)	$\Sigma(y)$ (m^2)	$\Sigma(\theta)$ (rad^2)	Particles
1	0.01472	0.01090	0.00104	165
2	0.01828	0.01121	0.00125	129
1∼2	0.01644	0.01105	0.00114	148

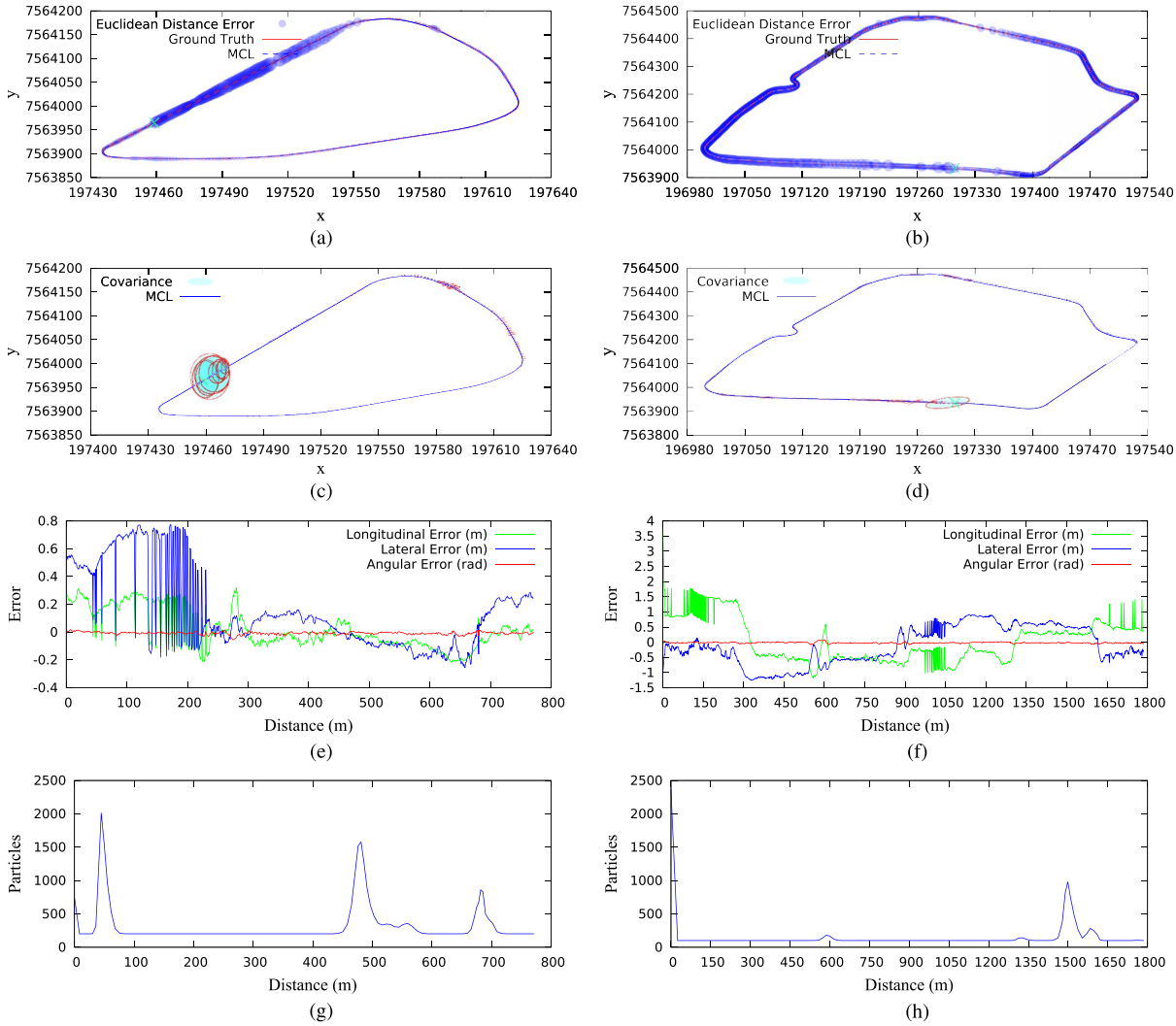


Fig. 11. MCL results obtained in the first lap of Tracks 1 and 2. (a) and (b) MCL positioning compared with ground truth. (c) and (d) MCL positioning covariance. (e) and (f) MCL positioning error along the trajectory. (g) and (h) Number of particles used along the street. (a) Track 1—Ground truth comparison. (b) Track 2—Ground truth comparison. (c) Track 1—MCL covariance. (d) Track 2—MCL covariance. (e) Track 1—Lateral, longitudinal and angular errors. (f) Track 2—Lateral, longitudinal and angular errors. (g) Track 1—Particle numbers. (h) Track 2—Particle numbers.

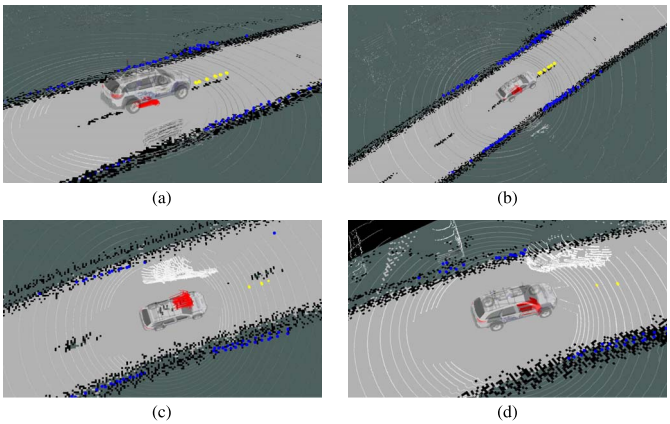


Fig. 12. (a) and (b) MCL performing in obstacle free roads. (c) and (d) MCL performing in roads with the presence of dynamic obstacles. Each red arrow represents one MCL particle position.

Fig. 12 shows the MCL performing in a road with and without dynamic obstacles (cars). The MCL particle position is represented by a red arrow. Particles show a slightly larger distribution when obstacles are handled.

In terms of processing time, our system spends an average of 0.10 s for a single pose estimation (i.e. works in a frequency of 10 Hz). This frequency is constrained by the frequency of the LIDAR sensor which is 10 Hz.

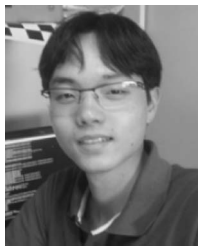
V. Conclusion

This paper reported on an autonomous car localization solution that uses road curbs and road marking as MCL features. In summary, the proposed curb detector can handle occluding situations by using LTS fitting. The novel road marking detector method is based on Otsu thresholding, which is insensitive to illumination. The robustness of the curb and road marking detector enabled the building of highly accurate environment maps and localization of a car even in the presence of other vehicles.

The localization experiments were performed in two test scenarios with lateral and longitudinal errors of less than 0.30 m. These results can be considered satisfactory for autonomous navigation in most urban scenarios, although we intend to investigate the use of other map representations and variations of Monte Carlo technique to further reduce the localization error. In order to minimize the localization error in long straight roads, the inclusion of other features are being studied and also the use of wheel encoders.

References

[1] L. Wei, C. Cappelle, and Y. Ruichek, "Horizontal/vertical LRFs and GIS maps aided vehicle localization in urban environment," in *Proc. IEEE 16th ITSC*, Oct. 2013, pp. 809–814.
[2] B. Qin *et al.*, "Curb-intersection feature based Monte Carlo localization on urban roads," in *Proc. IEEE ICRA*, 2012, pp. 2640–2646.
[3] D. Gruyer, R. Belaroussi, and M. Revilloud, "Map-aided localization with lateral perception," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2014, pp. 674–680.
[4] R. Matthaei, G. Bagschik, and M. Maurer, "Map-relative localization in lane-level maps for ADAS and autonomous driving," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2014, pp. 49–55.
[5] M. Schreiber, C. Knoppel, and U. Franke, "LaneLoc: Lane marking based localization using highly accurate maps," in *Proc. IEEE IV Symp.*, Jun. 2013, pp. 449–454.
[6] H. Deusch *et al.*, "Multi-sensor self-localization based on maximally stable extremal regions," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2014, pp. 555–560.
[7] J. Siegemund, U. Franke, and W. Förstner, "A temporal filter approach for detection and reconstruction of curbs and road surfaces based on conditional random fields," in *Proc. IEEE Intell. Veh. Symp.*, 2011, pp. 637–642.
[8] F. Oniga and S. Nedevschi, "Curb detection for driving assistance systems: A cubic spline-based approach," in *Proc. IEEE Intell. Veh. Symp.*, 2011, pp. 945–950.
[9] S. Kammel and B. Pitzer, "Lidar-based lane marker detection and mapping," in *Proc. IEEE Intell. Veh. Symp.*, Eindhoven, The Netherlands, 2008, pp. 1137–1142.
[10] M. Thuy and F. P. León, "Lane detection and tracking based on lidar data," *Metrol. Meas. Syst.*, vol. XVII, no. 3, pp. 311–322, 2010.
[11] M. Montemerlo *et al.*, "Junior: The Stanford entry in the urban challenge," *J. Field Robot.*, vol. 25, no. 9, pp. 569–597, Sep. 2008.
[12] P. J. Rousseeuw and K. Driessen, "Computing LTS regression for large data sets," *Data Mining Knowl. Discov.*, vol. 12, no. 1, pp. 29–45, Jan. 2006.
[13] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," in *Proc. IEEE ICRA*, May 2010, pp. 4372–4378.
[14] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man Cybern.*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.



Alberto Y. Hata received the M.S. degree in computer science in 2010 from the University of São Paulo, São Carlos, Brazil, where he is currently working toward the Ph.D. degree with the Institute of Mathematics and Computer Science. He has been working on mobile robotics and machine learning. His current research interests are autonomous car perception, Gaussian process, and robot localization.

Denis F. Wolf received the Ph.D. degree in computer science from the University of Southern California, Los Angeles, CA, USA, in 2006. He is an Associate Professor with the Department of Computer Systems, University of São Paulo (ICMC-USP), São Carlos, Brazil. He is currently the Director of the Mobile Robotics Laboratory, ICMC/USP, and a member of the Center for Robotics. He has published more than 50 journal and conference papers in the last years. His current research interests are mobile robotics, machine learning, computer vision, and embedded systems.