# Robust curb detection and vehicle localization in urban environments

**Conference Paper** · June 2014

DOI: 10.1109/IVS.2014.6856405

Alberto Yukinobu Hata
University of São Paulo
**21** PUBLICATIONS **325** CITATIONS
SEE PROFILE

Fernando Osorio
University of São Paulo
**184** PUBLICATIONS **1,208** CITATIONS
SEE PROFILE

Denis Wolf
University of São Paulo
**154** PUBLICATIONS **1,753** CITATIONS
SEE PROFILE

# Robust Curb Detection and Vehicle Localization in Urban Environments

Alberto Y. Hata[1], Fernando S. Osorio[1], and Denis F. Wolf[1]

*Abstract*— **Curb detection is an important capability for autonomous ground vehicles in urban environments. It is particularly useful for path planning and safe navigation. Another important task that can benefit from curb detection is localization, which is also a major requirement for self-driving cars. There are several approaches for identifying curbs using stereo cameras and 2D LIDARs in the literature. Stereo cameras depend on image pair matching methods to obtain depth information. Although 2D LIDARs being able to directly return this information, only few curb points can be detected using this sensor. In this work we propose the use of a 3D LIDAR which provides a dense point cloud and thus make possible to detect a larger extent of the curb. Our approach introduces the use of robust regression method named least trimmed squares (LTS) to deal with occluding scenes in contrast of temporal filters and spline fitting methods. We also used the curb detector as an input of a Monte Carlo localization algorithm, which is capable to estimate the pose of the vehicle without an accurate GPS sensor. We conducted experiments in urban environments to validate both the curb detector and the localization algorithm. Both method delivered successful results in different traffic situations and an average lateral localization error of $0.52655$ m in a $800$ m track.**

## I. Introduction

Curbs are structures usually present in urban traffic spaces that delimits the road boundaries. They provide important information to navigation and path planning of autonomous vehicles [1]. Moreover it has been demonstrated that curbs are an essential feature to vehicle localization, once the GPS signal reception is limited in urban places due to the presence of buildings and trees [2]. Fully automated ground vehicles like [3], [4], [5] uses a pre-built map of the environment and match the detected curb information against it to obtain accurate pose estimation, which is a fundamental capability to self-driving cars.

Due to the relevance of curbs for autonomous ground vehicles, this topic has been addressed in several works in the literature. The majority of existing curb detection methods uses stereo cameras for this task [1]. A direct approach for detecting curbs using these sensors is representing the 3D point cloud obtained in a digital elevation map (DEM), and then analyze the height variation in the map [6], [7]. Other methods use stixel (rectangular superpixel) structure to represent the stereo data and pre-filter large obstacles like cars and pedestrians [8]. For curb detection a region-of-interest (ROI) is defined to analyze the height and intensity variation of a stixel region. Later, neural networks are trained to classify the ROI either as curb or not. One disadvantage of stereo approaches is the lower accuracy compared to LIDAR sensors and the dependency of methods to compute images disparity to obtain the depth information, which can be a computational bottleneck for real time applications (running on general-purpose computers).

Curb detection methods that rely on LIDAR sensors do not suffer from this problem due to the possibility to access the point cloud from the raw measurements. Thus a height variation in the points can be directly evaluated and those that has a pre-determined value classified as curbs [4], [5]. Another way to detect curbs using LIDAR data is to estimate the road plane through random sample consensus (RANSAC) method and computing the normal of a bunch of points [9]. These points will be classified as curbs if the angle between the obtained normal and the surface is lower than a threshold.

As 2D LIDARs do not obtain dense point clouds at once, a limited number of points can be detected as curbs in each frame. On the other hand, 3D LIDAR sensor (e.g. multilayer LIDAR) can suppress the lack of data. There are few methods that use 3D LIDAR sensor data in the literature. In [3] curb like obstacles can be detected by analyzing the ring compression of a multilayer LIDAR and in [2] the angle between the LIDAR points and the beam angle was considered to detect curbs.

Independently of the sensor adopted in the curb detection task, they are all susceptible to occlusion when considered a typical urban environment. The most common approach to deal with occlusions is by fitting a model (e.g. spline) in the curb model and remove points that not match with the model. In [7] the curb data is temporally integrated through conditional random field (CRF) method. The CRF calculates the curb model parameters and is also integrated with a Kalman filter to predict the next parameters. The work of [6] temporally integrates the curb points and estimates the curve parameters through least squares cubic spline fitting method. In [9], a Kalman filter based algorithm is used to estimate the curb model. When the Kalman filter fails the user can manually adjust the curb model. Kalman filter is also employed in [4] to remove false positives.

This paper proposes a novel curb detection method based on 3D LIDAR. Our approach is an extension of obstacle detector proposed in [3] and we introduce a curb detector robust to false positives that works even in occluded scenes. We propose the use of least trimmed squares (LTS) to compute the curb curve parameters with the advantage of being able to automatically detect outliers. Experimental tests suggest that LTS generates better fitting results than the widely used smooth splines. Using this approach temporal integration is not necessary, allowing for improved computational efficiency.

[1] Mobile Robotics Laboratory, University of São Paulo (USP), São Carlos, São Paulo, Brazil, {hata, fosorio, denis}@icmc.usp.br

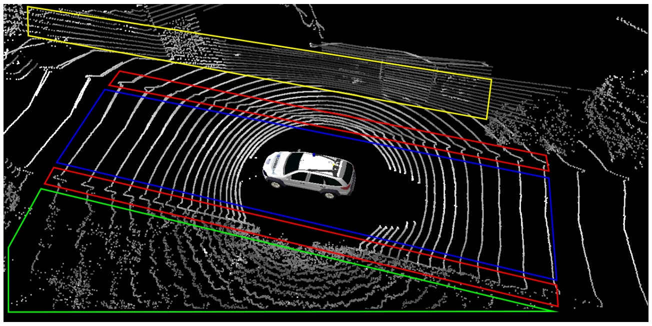


Fig. 1. Laser ring compression level is related to the surface slope. In this frame obstacles can be roughly distinguished as walls (yellow field), grass (green field), road (blue field) and curbs (red field). They are estimated using the ring compression data.

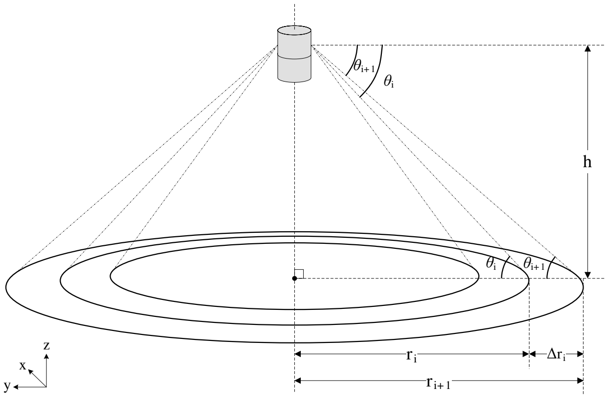


Fig. 2. Multilayer LIDAR rings intercepting a flat plane.

As previously stated, curbs can also be used as features for vehicle localization in urban streets. In this way, curb detection methods can be evaluated by applying in localization problems. The work of [5] integrated the 2D LIDAR curb detector with a Monte Carlo localization (MCL) method for autonomous navigation. The approach proposed by [2] matches the curb data with environment map in a probabilistic way using a 2D histogram filter. Similarly as [5], in this paper the detection method was integrated with MCL method in order to validate its applicability in urban scenes. The experiments were performed in a real autonomous vehicle platform, named Carina II which is equipped with a multilayer LIDAR.

This paper is organized as follows. Section II describes the curb detection method developed in this work. Section III presents the integration of the curb detection method and the MCL method. Section IV details the experiments conducted and the results obtained. Finally Section V presents the conclusion and future works.

## II. Curb Detection

Multilayer laser sensors like the Velodyne HDL model LIDAR are composed by several laser emitters that return concentric rings for each measurement. As demonstrated in [3], obstacles can be detected using this sensor by analyzing the distance between consecutive rings. Considering an urban scene illustrated in Figure 1, this distance can vary according to the surface slope (e.g. walls have higher ring compression compared to the road). The first step of the curb detector is to analyze adjacent ring points and check if the distance between them is lower than a threshold. Points that satisfy this condition are pre-classified as curbs. After that, filters are applied to remove false positives and deal with occlusions. Thus the remaining points are considered curbs. Each step is described in the next subsections.

### A. Ring Compression Analysis

In order to determine which ring compression range characterizes curb obstacles, first we compute the regular distance between rings (i.e. the distance when the LIDAR lasers intercept a flat plane). Considering Figure 2, the ring radius is given by:

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

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

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

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

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

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

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

where $S_i$ is the range used to check if a point belonging to the $i$-th ring is a curb. The parameters $\alpha$ and $\beta$ adjusts the range bounds, with $\beta > \alpha$.

LIDAR points are stored in a circular grid with angular resolution of 10 degrees to simplify the comparison of points placed in consecutive rings. Each annulus $i$ of this grid stores points of the same ring. Each cell $c_{i,j}$ contains the mean value $(x_{i,j}, y_{i,j}, z_{i,j})$ of the points position that fall in and the mean euclidean distance $d_{i,j}$ of these points to the sensor (in the $xy$ plane reference). Based on this data structure, distances between adjacent rings can be 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 $i$-th ring and $d_{i,j}$ represents the distance to the sensor stored in the same cell. In this manner, cells that have value of $\Delta d_{i,j}$ within the range $S_i$ are classified as curb candidates.

### B. False Positive Filters

Analyzing points solely according to its distance to the next ring can cause misclassification of obstacles that have

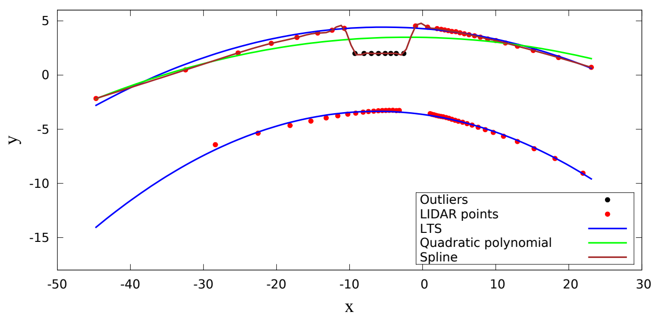


Fig. 3. Comparison of LTS with other curve fitting methods to estimate the curb shape.

| | SSE | R$^2$ |
|---|---|---|
| Least squares (quad. pol.) | 27.68531 | 0.59519 |
| Spline ($p = 0.99$) | 0.48798 | 0.99286 |
| LTS (quad. pol.) | 1.33002 | 0.97789 |

TABLE I

PERFORMANCE COMPARISON OF DIFFERENT FITTING METHODS TO ESTIMATE THE CURB MODEL.

same slope as curbs. Thus three filters are applied in the candidates set to remove false positives. In this paper these filters are named as: differential filter, distance filter and regression filter. These filters are applied in the following sequence.

*1) Differential Filter:* Curbs can be seen as urban structures with certain elevation relative to the road. Thus only points that have height variance relative to its neighboring may be classified as curbs. To preserve points with height variance we adapted the differential filter proposed in [4]. The differential filter is comprised by the operation of convolution over the height $(z)$ values of all cells of the circular grid. Here we used the following convolution mask: $\boxed{-2}\boxed{0}\boxed{+2}$ (using this mask in the cell $j$, the cells $j-1$ and $j+1$ are considered). The convolution result is verified if it is higher than a threshold $t_s$ in the curb candidates set.

*2) Distance Filter:* The distance filter is used to remove objects whose height is similar to curbs and the differential filter could not eliminate (e.g. grass, bushes and gravel). Considering a free road, curbs are usually the nearest obstacle of a vehicle sideways. Based on this fact, the distance filter searches the nearest points to the vehicle. The first step of this filter is to divide each ring into four quadrants. For each quadrant the closest point relative to the lateral axis $(y)$ is preserved. The following equation represents the distance filter applied in each ring's quadrant:

$$f_{dist}(c_{i,j}) = \arg\min_{c_{i,j}} |y_{i,j}| \tag{6}$$

where $c_{i,j}$ is the $j$-th cell of $i$-th ring and $y_{i,j}$ is its corresponding lateral position value. For each ring at maximum $4$ cells are preserved.

*3) Regression Filter:* When obstacles as pedestrians and cars are present in the street, the distance filter will possibly detect them as curbs. These obstacles can cause occlusion to the sensor and make difficult to identify actual curbs. The regression filter is introduced to estimate the curb shape and to remove points that do not follow the road model. The least trimmed squares (LTS) regression method is applied to fit a function in the curb candidate points. Compared to other fitting methods, LTS can compute the function parameters even in the presence of outliers. Figure 3 shows the curve estimation of a road data through three different fitting methods and its performance comparison presented in Table I. Despite of the spline fitting obtaining the best sum of squared errors of prediction (SSE) and the coefficient of determination (R$^2$), it considers the outliers as part of the curb. The LTS regression obtains a slightly lower performance than the spline, however it can obtain satisfactory curve models in the presence of outliers.

The LTS method is based on least squares regression theory with the advantage of computing the function parameters of the data subset that minimizes the cumulative error (i.e. the subset that does not contains outliers) [10]. For a dataset with $n$ points and $h$ the portion that does not contain outliers with $\frac{n}{2} < h < n$, the function parameters are obtained through the minimization of the following function:

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

where $r$ is the residual function and $\gamma$ is the curve function parameters. The curve model considered to compute the parameters was the quadratic polynomial. The curve parameters are calculated for both left and right road sides. Thus the curve shape is modeled after the resulting function and that points that are distant by a threshold $t_d$ are discarded. Finally the remaining points are classified as curbs.

### C. Parameter Tuning

In order to obtain satisfactory results in the curb detection we can exhaustively test several values for parameter set $\theta = \{\alpha, \beta, t_s, t_d\}$. However this approach can take a long time and is not guaranteed that the best parameters will be achieved. Thus we automatized the parameter search through a tool maned Meta-Evolver[1]. In this a objective function must be defined to establish the fitness of parameters. We adopted the following equation:

$$f_{tun}(\theta) = \arg\max_{\theta} \frac{n}{\text{MSE}}(\theta) \tag{8}$$

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 curb (after filtering process). The MSE is defined as:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(pos_i^{detc} - pos_i^{real})^2 \tag{9}$$

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

where $pos^{detc}$ refers to the detected curb point using the current parameter values and $pos^{real}$ is the actual curb point position. Thus the optimal situation is when there are a large number of points detected as curbs and the same time a low mean squared error.

The first step of Meta-Evolver is take a random walk in the parameter space based on arbitrary values of $\theta$. After obtaining some values of $\theta$ and the corresponding $f_{tun}$ a linear function is fitted in this values. The direction that the values must be explored is given by the obtained linear function. After reaching a number of iterations or when there is no change in the values of $\theta$, the search is stopped. Then the resulting parameters are used in the curb detector method.

## III. Vehicle Localization

For the vehicle localization we adopted the Monte Carlo localization (MCL) algorithm that relies on a map of the environment. In this work the environment is represented by an occupancy grid map (OGM) which stores the points obtained by the curb detector.

The MCL method is based on bayesian filter that estimates the position by matching the sensor measurements with the previously built map on the space. The position is given by a set of particles that is continuously updated after each sensor reading. Here the MCL was adapted to work with the curb detector. Thus only the detected curb points are considered as measurement features.

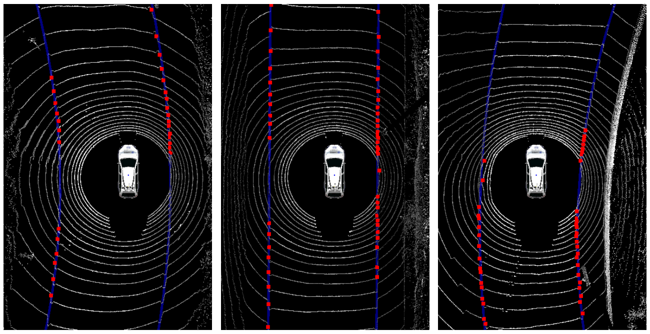
## IV. Results

For the experiments the CaRINA 2 test vehicle equipped with a Velodyne HDL-32E multilayer LIDAR has been used. The vehicle motion estimation has been obtained by a Xsens MTi-G GPS/IMU unit. However, no absolute position information has been used in the tests. The experiments were conducted in the following order: curb detection, environment mapping and vehicle localization. The following subsections details each experiment.

### A. Curb Detection

The first step of curb detection is to determine the best values for parameter set $\theta$ through parameter tuning. Three scenarios were considered to compute the objective function $f_{tun}$ (Equation 8): a straight road, a left turn and a right turn. For each scenario, curb points were manually marked to compute the $\mathrm{MSE}$ (Equation 9). The parameter search method has been limited to 500 iterations, and 20 executions of the method have been performed. The parameter set resulted from the execution with the highest $f_{tun}$ was used in the curb detection.

The best execution ($f_{tun} = 23033$) resulted in the parameter values: $\alpha = 0.113$, $\beta = 1.375$, $t_s = 0.124$ m and $t_d = 0.596$ m. Figure 4 presents results of the curb detection with the obtained parameters. The curb detection $\mathrm{MSE}$ of each scenario is $2.507 \times 10^{-4}$ $\mathrm{m}^2$ (left turn), $4.53 \times 10^{-3}$ $\mathrm{m}^2$ (straight road) and $1.532 \times 10^{-3}$ $\mathrm{m}^2$ (right turn). The curb detection has been tested with a large variety of curbs with positive results. Thus it is not necessary retrain the parameters unless the test environment is considerably different than typical urban scenarios.



(a) Left turn (b) Straight road (c) Right turn

Fig. 4. Scenarios employed in the parameter tunning. Red points are detected curbs and the blue line is the curb model estimated by LTS.

Figure 5(a)–(e) presents results of each step of curb detection. The scenes (a) and (b) represent a left curb occluded by a car and grass next to it, and a clean right curb. Employing ring compression results in incorrect detection of parts of the grass, paving and the car. Differential and distance filters (c) removes false positives of paving and grass, but still detects the car as curb. Finally, the regression filter (d) identifies the car as outlier and removes its points (e). Other examples of curb detection in occluding situations are presented in Figure 5(f) and (g). In the conduced experiments, the LTS regression could properly remove all false positives and also estimate the road shape in occluded scenes.

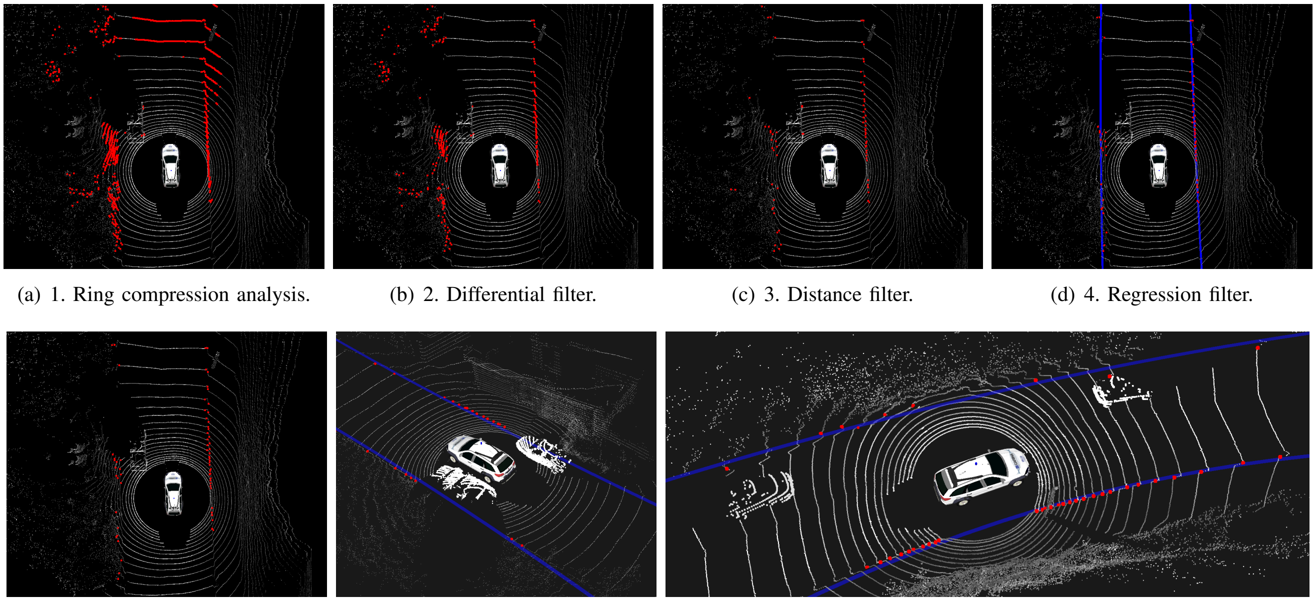
### B. Curb Maps

The map building was performed in a circuit of approximately 800 m presented in Figure 6. In the first moment we traversed the circuit (clockwise direction) and collected curb points positioned by GPS odometry. Later the curb points were registered through a ELCH offline SLAM method [11] and stored into a OGM with 0.10 m resolution. The resulting OGM is presented in Figure 6 and associated with the corresponding aerial image for comparison.

### C. Localization using Curb Features

The localization experiment has been performed in the same path as illustrated in Figure 6. Although the MCL provides full 2D pose estimation, our main focus is on the lateral localization, to ensure that the vehicle is in the correct traffic lane. As most the test environment is composed by straight (and simetric) paths, we expect an error in the longitudinal localization. In the initial setup, the map information and the points returned from curb detector were integrated to MCL method. The number of particles used in the experiment has been set to 100 (minimun) to 5000, which can be computed in real time even on a low end computer.

The MCL particles update is shown in Figure 8(a)–(b). In the initial instant (Figure 8(a)) particles are spread around the initial position. In this experiment, the position returned by the GPS was used as the initial position. Because of the



(a) 1. Ring compression analysis.
(b) 2. Differential filter.
(c) 3. Distance filter.
(d) 4. Regression filter.
(e) 5. Curb detection final result.
(f) Cars on both sides.
(g) Cars at front and behind.

Fig. 5. (a) – (e) curb detection steps. These scenes illustrate a car occluding the left curb. The curb detector identifies the presence of outliers and properly remove the false positives. (f) – (g) curb detection in other occluded scenes. Red points represents the curb points and blue lines represents the lane shape estimation using LTS regression.

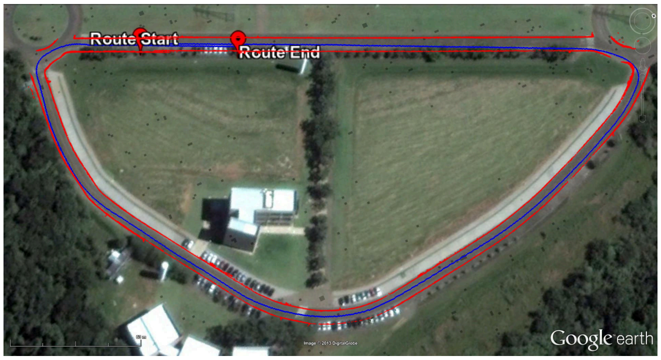


Fig. 6. Circuit used to perform mapping and localization experiments. Blue dots denote GPS raw points. Red dots denote the curb points and the resulting OGM.

| | $x$ (m) | $y$ (m) | $xy$ (m) | Var(x) ($\mathrm{m}^2$) | Var(y) ($\mathrm{m}^2$) |
|---|---|---|---|---|---|
| Mean | 1.317 | 0.582 | 1.495 | 0.189 | 0.097 |
| Std. Dev | 0.393 | 0.482 | 0.407 | 0.500 | 0.130 |

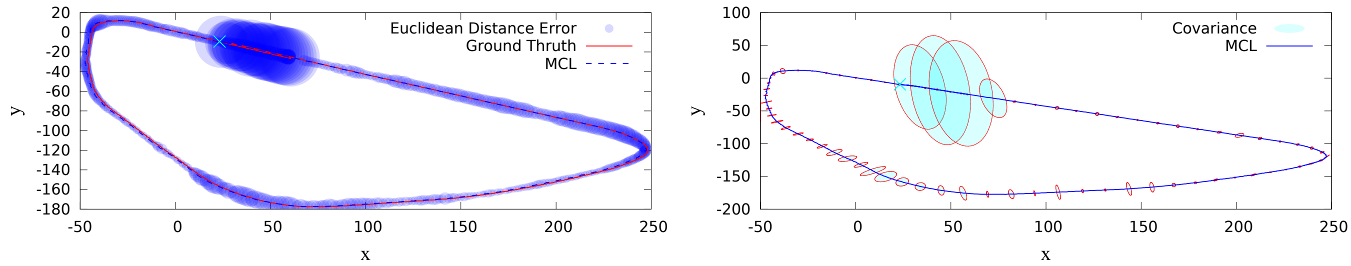
TABLE II
STATISTICS OF MCL EXPERIMENT. $x$, $y$ AND $xy$ CORRESPOND TO LONGITUDINAL, LATERAL AND ABSOLUTE EUCLIDEAN DISTANCE ERROR, RESPECTIVELY. Var(x) AND Var(y) CORRESPOND TO MCL LONGITUDINAL AND LATERAL ERROR VARIANCES, RESPECTIVELY.

GPS error, the MCL spends some iterations until the particle convergence. Figure 7(a) presents the Euclidean distance error between the estimated localization position and the ground truth (high precision GPS data). The localization starting point is denoted by light blue "×" and the circle's size is proportional to the error value. Due to the imprecise initial position returned by the GPS, the error is higher next to this point (large circles). In the first $42$ m the total euclidean error was around $7$ m.

The localization covariance error estimated also demonstrates the difficulty to match curb points with the environment map next to the initial position as shown in Figure 7(b). The ellipses $x$ and $y$ radius were exchanged by each other for a better illustration. After few meters the covariance value reduced considerably, which means that the curb points matched better with the map. Table II presents the statistics of the position error resulted from MCL localization after the convergence. This table shows that the longitudinal error was the major responsible for the localization error (mean $1.49$ m), despite of a relatively low lateral error of approximately $0.52$ m. Figure 9 illustrates the evolution of the error relative to the traversed distance after the MCL convergence. The systematic error is due to the longitudinal error. It is also important to mention that the curb is at the same height of the road in the area that corresponds to the bottom of the trajectory in Figure 6. In these cases they could not be detected, which increased the localization error in those areas, as it can be seen in Figure 7(b).

## V. Conclusion and Further Work

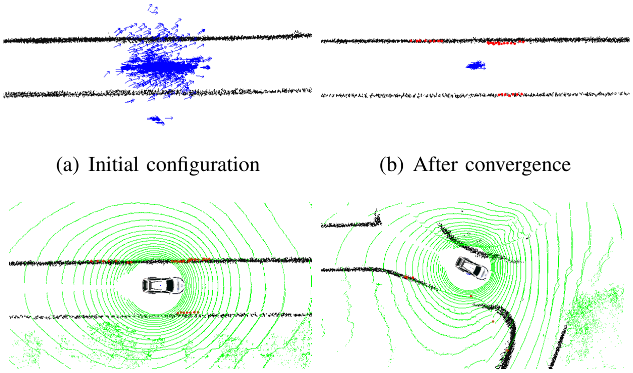
This paper presented a novel method for curb detection and its application for localization in urban scenarios. We used a multilayer LIDAR to gather environment data and extended the obstacle detection method proposed by [3] to extract curb structures. Filters were used to remove obstacles with same height variation as curbs, but the introduction of



(a) MCL position error

(b) MCL covariance error

Fig. 7. MCL results analysis. Errors are larger next to the initial position (denoted by light blue ×). (a) Circles representing the MCL position error compared to ground truth. (b) Ellipses representing the MCL covariance error. The ellipses $x$ and $y$ radius were exchanged by each other for a better illustration.



(a) Initial configuration

(b) After convergence

(c) Vehicle position in a straight road

(d) Vehicle position in the roundabout

Fig. 8. (a) and (b) MCL particles update. (c) and (d) position of the vehicle in the map estimated by MCL. Black, blue, red and green points represents map, particles, laser raw measurements and the detected curb points, respectively.

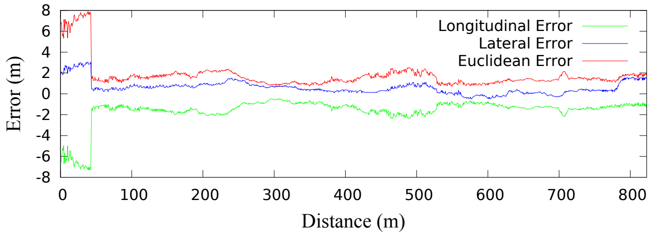


Fig. 9. MCL results analysis. Longitudinal, lateral and euclidean errors.

LTS robust regression was decisive to discard obstacles that caused occlusion. Thanks to the capacity of LTS fitting to deal with outliers dispensed the use of temporal integration and model prediction as Kalman filters. By performing fitting comparison between LTS and commonly used smooth splines in urban data demonstrated the superiority of the first one.

The developed detector was validated in a vehicle localization problem in order to ascertain its robustness in real urban environment. We built a high definition map of a $800\ \mathrm{m}$ circuit by registering the detected curb data. By performing the MCL in this circuit we obtained a lateral position error of $0.52\ \mathrm{m}$, including regions where the curb could not be detected.

For the future work we propose the integration of lane marking detection with the localization method in order to obtain a more accurate result and other urban scenes will also be considered in the next experiments.

## Acknowledgment

The authors acknowledge the grant provided by FAPESP (process no. 2012/02354-1) and thank the Mobile Robots Laboratory team for its support.

## References

[1] A. Bar Hillel, R. Lerner, D. Levi, and G. Raz, “Recent progress in road and lane detection: a survey,” *Machine Vision and Applications*, pp. 1–19, 2012.

[2] J. Levinson and S. Thrun, “Robust vehicle localization in urban environments using probabilistic maps,” in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, may 2010, pp. 4372 –4378.

[3] 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,” *Journal of Field Robotics*, vol. 25, no. 9, pp. 569–597, 2008.

[4] W. Zhang, “Lidar-based road and road-edge detection,” in *Intelligent Vehicles Symposium*, 2010, pp. 845–848.

[5] B. Qin, Z. J. Chong, T. Bandyopadhyay, M. H. Ang, E. Frazzoli, and D. Rus, “Curb-intersection feature based monte carlo localization on urban roads.” in *ICRA*. IEEE, 2012, pp. 2640–2646.

[6] F. Oniga and S. Nedevschi, “Curb detection for driving assistance systems: A cubic spline-based approach.” in *Intelligent Vehicles Symposium*. IEEE, 2011, pp. 945–950.

[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 *Intelligent Vehicles Symposium*. IEEE, 2011, pp. 637–642.

[8] M. Enzweiler, P. Greiner, C. Knöppel, and U. Franke, “Towards multicue urban curb recognition.” in *Intelligent Vehicles Symposium*. IEEE, 2013, pp. 902–907.

[9] A. Hervieu and B. Soheilian, “Road side detection and reconstruction using lidar sensor,” in *Intelligent Vehicles Symposium*, June 2013, pp. 1247–1252.

[10] P. J. Rousseeuw and K. Driessen, “Computing lts regression for large data sets,” *Data Min. Knowl. Discov.*, vol. 12, no. 1, pp. 29–45, Jan. 2006.

[11] J. Sprickerhof, A. Nüchter, K. Lingemann, and J. Hertzberg, “A heuristic loop closing technique for large-scale 6d slam,” *AUTOMATIKA – Journal for Control, Measurement, Electronics, Computing and Communications, Special Issue with selected papers from the 4th European Conference on Mobile Robots*, vol. 52, no. 3, December 2011.