

Monte Carlo Localization on Gaussian Process Occupancy Maps for Urban Environments

Alberto Y. Hata[✉], Fabio T. Ramos, and Denis F. Wolf

Abstract—Map-aided localization methods have been employed for vehicle localization to overcome the limitations of global navigation satellite system (GNSS) devices. In this solution, sensor information is matched to the environment map to determine the vehicle position. Occupancy grid maps (OGMs) have been adapted for map-aided localization. However, there are known drawbacks of OGM, such as the environment discretization, the assumption of independence between grid cells, and the need for dense measurements. In recent years, Gaussian process occupancy map (GPOM) was developed to suppress some of the OGM limitations. GPOM enables the computation of the likelihood of occupancy at any location, even if not directly observed by the sensor, thus representing the environment in a continuous manner. Taking into account the superiority of GPOM over OGM, we devise a novel vehicle localization technique for urban environments. This solution enables more accurate localization due to the use of a representation that better models the real environment. The development of the proposed method is based on Monte Carlo localization, which is a popular map-aided localization method. Two road features commonly found in urban cities were chosen to build the maps: road curbs and road markings. Specifically, the proposed localization method relies on a GPOM constructed with curb data and an OGM built with road marking data. Experiments were performed in real urban environments. Maps were intentionally generated using sparse light detection and ranging (LIDAR) data to verify the localization in non-observed areas. The localization system was evaluated by comparing the results with a high precision GNSS device.

Index Terms—Vehicle localization, Gaussian process occupancy map, occupancy grid map, curb detection, road marking detection, Monte Carlo localization.

I. INTRODUCTION

THE autonomous vehicles are comprised of fundamental systems, such as obstacle detection, navigation, and localization [1]. Localization plays an important role as it determines the vehicle position in the environment. Based on

Manuscript received December 11, 2016; revised June 25, 2017 and September 8, 2017; accepted October 3, 2017. This work was supported in part by FAPESP under Grant 2012/02354-1 and Grant 2014/09096-3, in part by the School of Information Technologies members, and in part by Mobile Robots Laboratory members. The Associate Editor for this paper was S. S. Nedevschi. (*Corresponding author: Alberto Y. Hata*)

A. Y. Hata and D. F. Wolf are with the Mobile Robotics Laboratory, University of São Paulo, São Carlos 13566-590, Brazil (e-mail: alberto.hata@usp.br; denis@icmc.usp.br).

F. T. Ramos is with the School of Information Technologies, The University of Sydney, Sydney 2006, Australia (e-mail: fabio.ramos@sydney.edu.au).

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.2017.2761774

localization information, we can compute the position of an obstacle and perform the path planning.

Despite the practical solution for vehicle localization is the use of high precision Global Navigation Satellite System (GNSS) devices, there are external factors that hamper the reception of satellite signals inside cities (e.g. urban canyons) which can decrease the position accuracy or even making it unusable [2]. To overcome the limitations of the GNSS devices, environment maps created beforehand are employed to aid during the vehicle localization. Through this process, the position is estimated even in the absence of satellite

Despite the extensive use of OGM for map-aided localization [3], [4], [7], there are some disadvantages associated with this structure. First, OGM requires the discretization of the environment into grid cells. Another drawback is the assumption of independence between grid cells, thus the neighbor cells are not taken into account when building the map [5].

Recently, O'Callaghan and Ramos [5] developed a novel metric map structure named Gaussian process occupancy map (GPOM). This map is modeled through a machine learning method called Gaussian process (GP) [6]. Thanks for the GP prediction robustness, non-observed areas of the scenario are estimated from other sensor measurements and thus a dependent model is employed. Additionally, GPOM discards the environment discretization, therefore, space is represented in a continuous way.

Given the advantages of GPOM, its application in vehicle localization problems may increase the accuracy of the estimated pose, once a more reliable representation of the environment is taken into account. Moreover, the estimation of the occupancy of unseen areas by GPOM makes it possible to localize in such areas that was not previously observed. The proposed localization method is based on Monte Carlo localization (MCL) algorithm, because of its capability to deal with ambiguous situations and noisy sensor readings. We devise a novel measurement likelihood function to calculate the particle weights, given the detected curbs and the GPOM. Particularly, the likelihood is calculated by feeding the free and the occupied space information obtained from light detection and ranging (LIDAR) measurements in the multivariate normal probability density function (MNPDF).

As stated before, OGM representation is a common choice for map-aided vehicle localization. Generally, MCL is employed for OGM based localization. The work of [3] employs MCL with likelihood field laser model to obtain the likelihood of a detected curb to match the OGM.

The likelihood field values of the map is computed beforehand to reduce the computing complexity. In [7], MNPDF is used to match detected curbs and road markings to an urban map. Following this idea, [8] proposed the use of probability density function to calculate the measurement likelihood that includes road marking and curb data. The main drawback of these approaches is the dependence on a dense map which is obtained by repetitive measurements of the same place using high-resolution sensors. Some works also employ map matching instead of MCL as in [9] and [10], but it is less tolerant to noises than MCL and unable to recover from localization failures. Besides OGM, other authors employ third party maps which are represented by vector graphics and refined through manual adjustments. In [11], transversal planes of the raw LIDAR point cloud is matched to a Geographic Information System (GIS) map through Iterative Closest Point algorithm. The GIS map is also used in [12], but the measurement likelihood of the detected curb and road marking features are calculated through MNPDF. Similarly, [13] employed a high definition map of the environment built from a specialized company and integrated with a MCL. Despite GIS maps provide accurate models of the environment, they are not publicly available and may not provide geographic information of any location. Regards to GP, it has already been studied to enhance the localization, but in a different context. In [14] GP was used to localize a robot using WiFi signal and in [15] a dense LIDAR measurement was generated through GP to improve MCL robustness.

Differently from these approaches, we propose a novel vehicle localization method that integrates the GPOM representation with MCL thus addressing OGM limitations and producing high-resolution details of the environment as GIS maps. Our approach enables accurate localization in partially observed areas during the mapping stage and, in contrast to methods that depends on fusion of several sensors for data redundancy [10], it supports sparse or incomplete measurements gathered from a single sensor. The main contributions of this paper are: (a) a localization method that relies on a continuous representation to estimate the occupancy of unseen areas (GPOM); (b) accurate localization in regions that were not observed during the mapping stage; (c) a technique for overall mapping and localization performed with a low resolution and inexpensive sensor; and (d) the formulation of a new MCL likelihood function based on MNPDF that takes into account the uncertainty of the occupancy predictions from GPOM.

The remaining paper is organized as follows: Section II describes the road reconstruction through Gaussian process occupancy mapping, as well as the strategies to represent large areas; Section III describes the proposed localization method and the measurement likelihood model that supports GPOM; and finally Section IV describes the performed experiments in real urban streets and the obtained results.

II. GAUSSIAN PROCESS OCCUPANCY MAP

The GPOM is based on a supervised machine learning method named Gaussian process (GP). The mapping process estimates the occupancy of any point by obtaining the

relationship between the sensor measurements. In this work, the GPOM is constructed from the detected road features.

A. Road Feature Detection

Urban maps are composed of environmental structures that are extracted from sensor measurements. Here, we focus on the detection of two road structures: curbs and road markings. These features are used to build the urban maps and later during the localization. Precisely, the GPOMs were constructed based on curb data and additionally, they can be combined OGMs built with road markings. Details of curb and road marking detection methods are described in [16].

B. Gaussian Process

The central idea of GP is to explore the correlation between input and its corresponding target values through a given kernel (covariance function) [6]. GP is generally employed in regression problems with the advantage of having a fast learning process and being tolerant to noise in the training dataset [17]. Differently from other regression methods, GP does not require a prior definition of the basis function of the curve that models the dataset (*i.e.* non-parametric).

More precisely, a GP consists of a multivariate Gaussian distribution modeling the function space of a dataset $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^N$, where $\mathbf{x}_i \in \mathbb{R}^D$ and $y_i \in \mathbb{R}$. The model is fully specified by a mean function $\mu(\mathbf{x})$, and a covariance function $k(\mathbf{x}, \mathbf{x}')$. The covariance function is also known as kernel.

The GP prior of a query data \mathbf{x}_* (*i.e.* the test value) is a Gaussian distribution with mean 0 and variance given by the covariance of the query point as follows:

$$y_* = \mathcal{N}(0, k(\mathbf{x}_*, \mathbf{x}_*)). \quad (1)$$

Given the GP prior and conditioning on the training dataset \mathcal{D} , the posterior y_* is also a Gaussian:

$$p(y_* | \mathcal{D}, \mathbf{x}_*) = \mathcal{N}(\mu(\mathbf{x}_*), \sigma(\mathbf{x}_*)). \quad (2)$$

The mean and the variance of the posterior are represented respectively by $\mu(\mathbf{x}_*)$ and $\sigma(\mathbf{x}_*)$, which are obtained by:

$$\mu(\mathbf{x}_*) = k(\mathbf{x}_*, \mathbf{x})^T [k(\mathbf{x}, \mathbf{x}) + \sigma_n^2 I]^{-1} y, \quad (3)$$

$$\sigma(\mathbf{x}_*) = k(\mathbf{x}_*, \mathbf{x}_*) - k(\mathbf{x}_*, \mathbf{x}) [k(\mathbf{x}, \mathbf{x}) + \sigma_n^2 I]^{-1} k(\mathbf{x}_*, \mathbf{x}), \quad (4)$$

where σ_n is the global noise value of the dataset \mathcal{D} . The Gaussian Process of \mathbf{x}_* is also denoted as $\mathcal{GP}(\mu(\mathbf{x}_*), \sigma(\mathbf{x}_*))$. Detailed explanation about GP theory can be found in [6] and [17].

C. Gaussian Process Mapping

A single LIDAR beam is geometrically equivalent to a line with a starting and ending point. The end point is where the beam hit an object and the extent between start and end points is the space free of obstacles. Therefore, two types of observation arise from a single beam with a valid measurement: a point representing the occupied area and a line representing the free space. Hence, values of the dataset \mathcal{D} are

expressed as $\mathbf{x}_i \in \{x; l(u, v)|x, u, v \in \mathbb{R}^2\}$, where x denotes a point observation and $l(u, v)$ a line observation with u and v as the line boundary points.

Changing the mean and variance formulations of Equation 3 and Equation 4 to the mapping context, \mathbf{x}_i corresponds to the LIDAR beam positions and $y_i \in [0, 1]$ to the occupancy values of \mathbf{x}_i , specifically, 0 for free and 1 for occupied location. To predict the occupancy at \mathbf{x}_i , GP results are squashed into $[0, 1]$ interval through a sigmoid function. Details of the GP mapping method can be found in [5] and [19].

D. Information Theoretic Compression of Training Data

When gathering sensor data, redundant measurements are inevitably collected which could be eliminated for computing speed up. The unnecessary data can be discarded by adapting the information-theoretic compression of LIDAR data proposed by [18]. The idea behind the compression is to evaluate the mutual information of a measurement and the current dataset. In other words, only those readings that reduce the uncertainty about the environment are stored. Formally, the mutual information of the measurement set Z and the map m which represents the environment is given by:

$$I(m; Z) = H(m) - H(m|Z), \quad (5)$$

where I denotes the mutual information and H the entropy. Here m is an occupancy grid map which assumes independence between measurements.

To reduce the uncertainty, we must obtain the subset Z^* of Z in which:

$$Z^* = \underset{Z \subseteq Z_{1..n}, |Z| \leq n}{\operatorname{argmax}} H(m) - H(m|Z), \quad (6)$$

where n is the number of measurements.

The conditional entropy $H(m|Z)$ is given by the mean value of the entropy of each single measurement \mathbf{z} from Z :

$$H(m|Z) = \int_z p(\mathbf{z}) H(m|\mathbf{z}) d\mathbf{z}. \quad (7)$$

From this, the conditional entropy of a single measurement can be calculated by:

$$H(m|\mathbf{z}) = \sum_{C \in m} H(C|\mathbf{z}) = - \sum_{C \in m} \sum_{c \in C} p(c|\mathbf{z}) \log p(c|\mathbf{z}), \quad (8)$$

where C denotes a set of cells and c a single cell from map m . The value of the occupancy $p(c|\mathbf{z})$ can be retrieved through the inverse sensor model method.

In order to simplify the compression, we calculate $H(m|\mathbf{z})$ of all measurements Z and remove those that results in a conditional entropy lower than the average $H(m|Z)$. Here we consider each LIDAR beam as a single measurement \mathbf{z} . Therefore the resulting subset Z^* is formed by measurements that satisfy the following condition:

$$Z^* = \{\mathbf{z} \in Z \mid H(m|\mathbf{z}) \leq H(m|Z)\}. \quad (9)$$

E. Mixture of Gaussian Processes

Instead of generating one single map, the environment can be split into smaller regions and then produced a set of GPOMs. The strategy of using several GPs is named mixture of GPs and its application for mapping was proposed by [19].

The first step of the mixture of GPOMs is to cluster the measurement data according to certain criteria. In the current work, k -means clustering method was applied. Given a dataset with N sensor measurements and the maximum number of measurements S that each GP must handle, the number of clusters (GP mixtures) κ is set as.:

$$\kappa \geq \frac{N}{S}. \quad (10)$$

After clustering, for each measurement subset $\{z_i\}_{i=1}^\kappa$, a corresponding centroid $\{c_i\}_{i=1}^\kappa$ and a unique GP expert $\{\varepsilon_i\}_{i=1}^\kappa$ is associated with it.

When building the environment map, a set of test points $T = \{d_j\}_{j=1}^M$ must be evaluated. A gating network evaluates which expert should be chosen to infer the occupancy of the test points. For a test point d_j , we associate the expert ε_i whose corresponding cluster c_i is the closest to this point.

III. MONTE CARLO LOCALIZATION WITH GPOM

Monte Carlo localization (also known as particle filter localization) is based on Bayes filter and importance sampling theories to estimate the robot position from sensor measurements and environment map. The most likely robot poses are represented by a set of particles that is iteratively updated as new sensor observations are obtained [14]. In each iteration, those particles that produce a better matching of the measurement with the map are maintained in the set. Occupancy grid maps are commonly used, therefore adaptations are necessary to support GPOM.

In a probabilistic way, MCL estimates the probability density function (PDF) of the posterior of $p(\mathbf{x}_k|\mathbf{z}_{1:k}, m)$, where the robot position is represented by $\mathbf{x}_k = \{x_k, y_k, \theta_k\}$ (2D pose and orientation), the robot measurement is represented by \mathbf{z}_k , the time stamp as k and m as the environment map.

A. General MCL Algorithm

The MCL algorithm starts by randomly distributing particles over the environment. The density of $p(\mathbf{x}_k|\mathbf{z}_{1:k}, m)$ is represented by the particle set $\mathbf{S}_k = \{s_k^i; i = 1, 2, \dots, n\}$, 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 position certainty.

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

- 1) **Prediction step:** The motion update step estimates the position x_k^i of particles s_k^i . The position is calculated by the probability $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_{k-1})$, where \mathbf{x} and \mathbf{u} correspond to the robot position and the robot motion, respectively. The motion may be obtained by wheel encoder, IMU or GPS sensor. The estimated density is

given as:

$$\hat{p}(\mathbf{x}_k | \mathbf{z}_{1:k-1}, m) = \sum_{i=1}^n p(\mathbf{x}_k | s_k^i, \mathbf{u}_{k-1}). \quad (11)$$

- 2) **Update step:** The measurement update step incorporates the sensor data \mathbf{z}_k and map m to calculate the posterior density $\hat{p}(\mathbf{x}_k | \mathbf{z}_k)$. This step calculates the likelihood of s_k^i to generate the \mathbf{z}_k given m :

$$\hat{p}(\mathbf{x}_k | \mathbf{z}_{1:k}, m) \propto p(\mathbf{z}_k | \mathbf{x}_k, m) \hat{p}(\mathbf{x}_k | \mathbf{z}_{1:k-1}, m). \quad (12)$$

Importance sampling approximates an unknown distribution $f(x)$ from an approximated distribution $g(x)$ by weighting each sample with the term $w = \frac{f(x)}{g(x)}$. Using importance sampling to estimate the density we have:

$$p(\mathbf{x}_k | \mathbf{z}_{1:k}, m) = \frac{p(\mathbf{z}_k | \mathbf{x}_k, m) \hat{p}(\mathbf{x}_k | \mathbf{z}_{1:k-1}, m)}{\hat{p}(\mathbf{x}_k | \mathbf{z}_{1:k-1}, m)} \quad (13)$$

$$= p(\mathbf{z}_k | \mathbf{x}_k, m). \quad (14)$$

To simplify, this term is expressed as:

$$w_k^i = p(\mathbf{z}_k | \mathbf{x}_k, m), \quad (15)$$

where, w_k^i is the weight of the particle s_k^i .

- 3) **Resampling step:** n particles are chosen randomly from \mathbf{S}'_k (with replacement). Those particles with higher weights have proportional probability to be selected. The chosen particles replace the current \mathbf{S}_k set.

B. Likelihood Function for GPOM

In our measurement likelihood function, occupancy and geometric information are used. First, to get the occupancy information, LIDAR end point poses are obtained for each particle s_k^i , given its position $(x_k^i, y_k^i, \theta_k^i)$ and the current measurement $\mathbf{z}_k = \{(r_k^j, a_k^j), j = 1, 2, \dots, l\}$:

$$\mathbf{p}_k^i = \{p_k^{i,j}; j = 1, 2, \dots, n\}, \quad (16)$$

$$p_k^{i,j} = (x_k^i + r_k^j \cos(a_k^j + \theta_k^i), y_k^i + r_k^j \sin(a_k^j + \theta_k^i)) \quad (17)$$

where r_k^j and a_k^j denote the measurement \mathbf{z}_k range distance and beam angle of beam index j , respectively, and l is the number of beams. We highlight that i corresponds to the particle index and $p_k^{i,j}$ is the end point position of the beam j .

From the list of LIDAR end point poses \mathbf{p}_k^i , occupancy mean and variance are obtained from each LIDAR beam end point through GP prediction:

$$\text{occupancy}_k^i = \begin{bmatrix} \mathcal{GP}(\mu(p_k^{i,1}), \sigma(p_k^{i,1})) \\ \mathcal{GP}(\mu(p_k^{i,2}), \sigma(p_k^{i,2})) \\ \vdots \\ \mathcal{GP}(\mu(p_k^{i,l}), \sigma(p_k^{i,l})) \end{bmatrix}, \quad (18)$$

$$\mathbf{m}_k^i = \begin{bmatrix} \mu(p_k^{i,1}) \\ \mu(p_k^{i,2}) \\ \vdots \\ \mu(p_k^{i,l}) \end{bmatrix}, \quad \mathbf{v}_k^i = \begin{bmatrix} \sigma(p_k^{i,1}) \\ \sigma(p_k^{i,2}) \\ \vdots \\ \sigma(p_k^{i,l}) \end{bmatrix}, \quad (19)$$

where \mathbf{m}_k^i and \mathbf{v}_k^i corresponds to the mean and variance values obtained from \mathbf{p}_k^i .

Using solely occupancy information can easily lead to ambiguous position estimation as it results in particles headed to any obstacle. Therefore, the geometric information of the LIDAR measurements has been included into the likelihood model. We used the distance \mathbf{d}_k^i between the particle to the first point along the LIDAR beam direction which has occupancy mean higher than 0.99. Essentially, a LIDAR measurement is simulated from the particle pose to determine how far the particle is to the obstacle. For this, ray-casting is employed in each LIDAR beam.

$$\mathbf{d}_k^i = \begin{bmatrix} d_k^{i,1} \\ d_k^{i,2} \\ \vdots \\ d_k^{i,l} \end{bmatrix}, \quad (20)$$

where, $d_k^{i,j}$ is the distance from s_k^i pose to the closest obstacle along the direction given by beam j .

From this, it is possible to obtain the error ϵ_k^i between the measured range and the distance \mathbf{d}_k^i in terms of squared difference:

$$\boldsymbol{\epsilon}_k^i = \begin{bmatrix} (r_k^1 - d_k^{i,1})^2 \\ (r_k^2 - d_k^{i,2})^2 \\ \vdots \\ (r_k^l - d_k^{i,l})^2 \end{bmatrix} = \begin{bmatrix} \epsilon_k^{i,1} \\ \epsilon_k^{i,2} \\ \vdots \\ \epsilon_k^{i,l} \end{bmatrix}. \quad (21)$$

Two arrays are formed by combining the occupancy and the distance information associated with particle s_k^i :

$$\boldsymbol{\mu}_k^i = \begin{bmatrix} \mathbf{m}_k^i \\ \mathbf{d}_k^i \end{bmatrix}, \quad \boldsymbol{\sigma}_k^i = \begin{bmatrix} \mathbf{v}_k^i \\ \boldsymbol{\epsilon}_k^i \end{bmatrix}, \quad (22)$$

where $\boldsymbol{\mu}_k^i$ and $\boldsymbol{\sigma}_k^i$ consolidates the mean and variance values extracted from the measurement \mathbf{z}_k .

Finally, to calculate the particle weight w_k^i , we estimate the likelihood to obtain the values of $\boldsymbol{\mu}_k^i$ from the distribution given by the measurement \mathbf{z}_k . This is done by connecting the measurement and the particle information in the multivariate normal probability density function:

$$w_k^i = p(\mathbf{z}_k | \mathbf{x}_k^i, m) \quad (23)$$

$$= p(\mathbf{z}'_k | \mathbf{x}_k^i, \boldsymbol{\mu}_k^i, \boldsymbol{\Sigma}_k^i) \quad (24)$$

$$= \frac{1}{2\sqrt{\pi}|\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\left(\frac{1}{2}(\mathbf{z}'_k - \boldsymbol{\mu}_k^i)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{z}'_k - \boldsymbol{\mu}_k^i)\right), \quad (25)$$

where, $\boldsymbol{\Sigma}_k^i$ is the covariance matrix with $\boldsymbol{\sigma}_k^i$ as the diagonal values and \mathbf{z}'_k contains measurement values extracted from \mathbf{z}_k . The matrix representation of $\boldsymbol{\Sigma}_k^i$ and \mathbf{z}'_k is given as follows:

$$\mathbf{z}'_k = \begin{bmatrix} \mathbf{U}_l \\ \mathbf{r}_k^i \end{bmatrix}, \quad \boldsymbol{\Sigma}_k^i = \text{diag}(\mathbf{v}_k^i), \quad (26)$$

with $\mathbf{U}_l = [1, 1, \dots, 1]^T$ as a unitary array with l rows and $\mathbf{r}_k^i = \{r_k^j, j = 1, 2, \dots, l\}$ as the range values. The unitary array represents the occupancy values of the end points of \mathbf{z}_k (i.e. all end points fall in an occupied area).

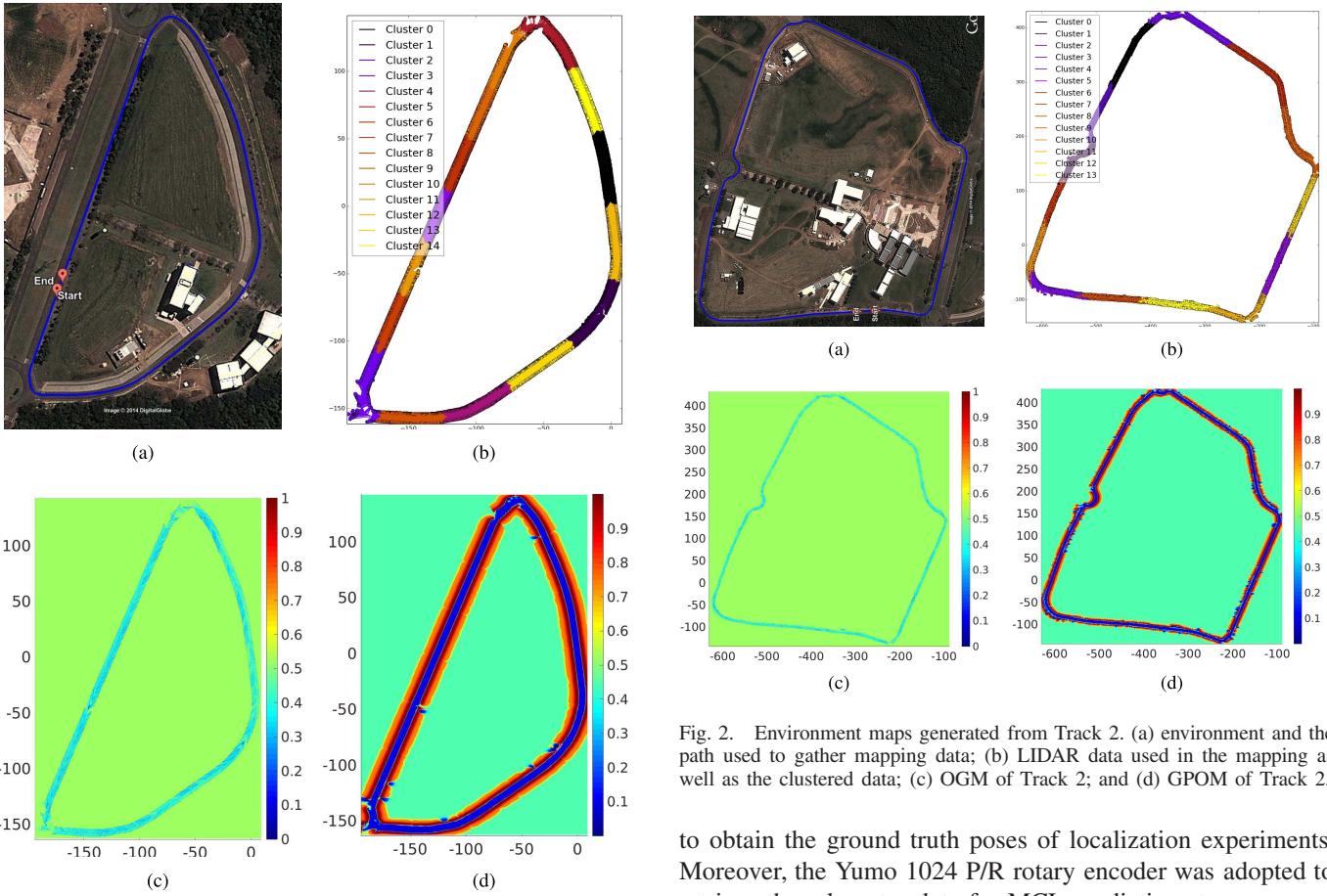


Fig. 1. Environment maps generated from Track 1. (a) environment and the path used to gather mapping data; (b) LIDAR data used in the mapping as well as the clustered data; (c) OGM of Track 1; and (d) GPOM of Track 1.

C. Likelihood for Multiple Map Sources

The proposed MCL uses GPOM built from curb data, but also supports OGM constructed with road marking data for a more accurate localization. To integrate both maps, curb and road marking observations are processed separately. The likelihood of curb observations are calculated through Equation 25 and the likelihood of road marking observations are evaluated through the LIDAR likelihood field model. The MCL likelihood equation that takes into account both measurements is given as:

$$w_k^i = p(\mathbf{z}_k^{\text{curb}} | \mathbf{x}_k^i, m_{\text{gpom}}) \times p(\mathbf{z}_k^{\text{rmark}} | \mathbf{x}_k^i, m_{\text{ogm}}), \quad (27)$$

where $\mathbf{z}_k^{\text{curb}}$, $\mathbf{z}_k^{\text{rmark}}$, m_{gpom} and m_{ogm} represent the curb observation, road marking observation, OGM and GPOM, respectively.

IV. EXPERIMENTS AND RESULTS

Experiments were performed in real urban environments using the Carina II autonomous vehicle prototype [1]. For environment perception, a Velodyne HDL-32E multilayer LIDAR sensor was employed. This sensor returns a 3D point cloud with intensity values in each measurement. In mapping experiments, a Septentrio AsteRx2eH PRO was used to obtain the GNSS-RTK measurements. This device was further used

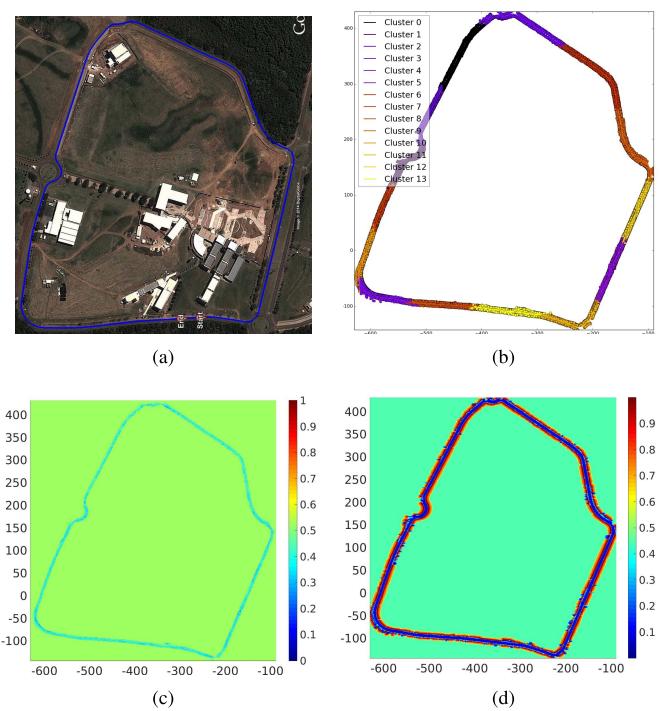


Fig. 2. Environment maps generated from Track 2. (a) environment and the path used to gather mapping data; (b) LIDAR data used in the mapping as well as the clustered data; (c) OGM of Track 2; and (d) GPOM of Track 2.

to obtain the ground truth poses of localization experiments. Moreover, the Yumo 1024 P/R rotary encoder was adopted to retrieve the odometry data for MCL prediction step.

Here the combination of MCL with GPOM and the combination MCL with OGM are denoted as MCL-GPOM and MCL-OGM, respectively. The following sections describe the mapping and localization experiments and the obtained results. The MCL-GPOM results are also compared to the MCL-OGM localization approach.

A. Environment Mapping

Resulting curb and road marking point clouds obtained from the detection methods were georeferenced (through GNSS-RTK) before mapping. From this, curb data was employed to build GPOM and road marking data to build OGM.

Two datasets were employed for mapping experiments, referred here as Track 1 and Track 2. These datasets consist of sensor data collected from streets inside the Campus 2 of USP São Carlos. Though each dataset contains sensor log of three laps, just one was used for mapping. The path covered by each dataset is denoted by the blue lines in Figure 1(a) and Figure 2(a). Track 1 is a 770 m long path and Track 2 is a 1749 m long path. Measurements were gathered during low traffic and the vehicle run at approximately 40 km/h.

Experiments and results of curb mapping and road marking mapping are presented in the following subsections.

1) Curb Mapping: Before mapping, 3D point clouds of the detected curbs are converted into 2D LIDAR range data to extract free and occupied space information. This processes is also necessary to generate 2D top view representations of

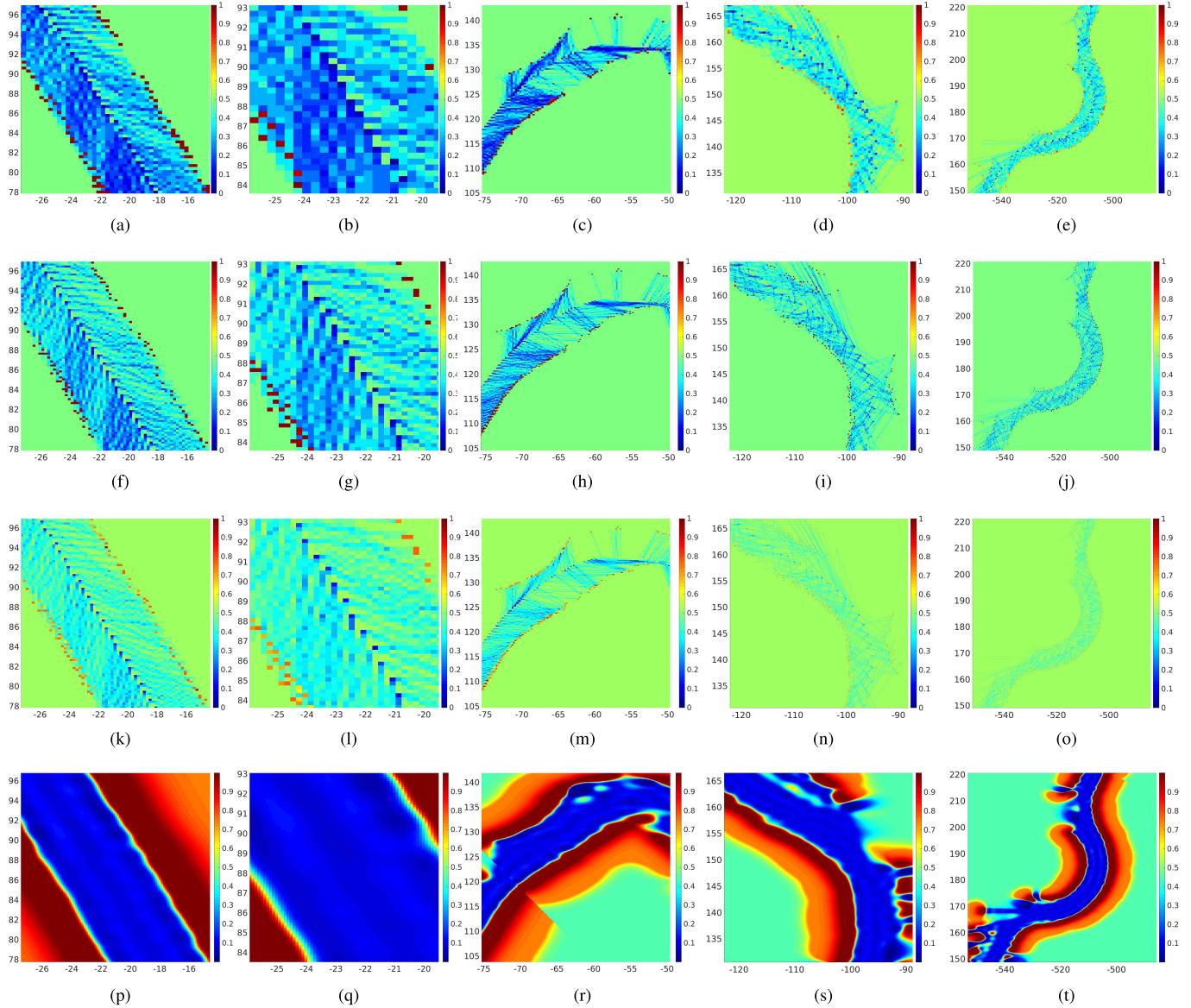


Fig. 3. Amplified regions of maps generated from Track 1 (A, B and C) and Track 2 (D and E). Blue and red colors are associated with free and occupied areas, respectively.

the environment (localization is performed on 2D maps). For this, it is simulated in the detected 3D curb points, a 360° LIDAR scan parallel to the ground and configured to return at maximum 90 equally spaced LIDAR beams (45 directed forward and 45 directed backward, corresponding to 1/4 of conventional LIDAR scan sensor resolution). Therefore, this conversion process results in up to 90 valid range values sampled from the curb points. The set of LIDAR scan data generated from curb points was georeferenced to form the mapping dataset. This dataset was then reduced through information theoretic data compression method to remove redundant measurements (subsection II-D). The resulting training dataset for Track 1 and Track 2 is formed by 17252 (39.88% reduction) and 11830 (40.03% reduction) LIDAR beams (*i.e.* lines and points), respectively. Training data measurements are illustrated in Figure 1(b) and Figure 2(b).

It is important to notice that the curb detection itself returns a sparse point cloud and it is additionally reduced to induce

the production of unseen areas. As the set of measurements of a single loop is used, the resulting mapping dataset is considerably sparse if considered the track extent.

The GP classifier kernel was set with squared exponential and the hyper-parameters were trained with gradient descent method. To reduce the computational cost, the training dataset was clustered to form 15 GP experts. All GP experts use the same hyper-parameters for occupancy inference.

To build the GPOM, test points must be provided to the GP model to obtain the GP mean and variance of the environment space. For this, a test dataset containing evenly spaced (x, y) points with 0.10 cm spacing that covers all the scenario was used. The occupancy of each point is inferred by the closest GP expert. The dataset portion associated to each GP expert is represented by a distinct color in Figure 1(b) and Figure 2(b). The inferred mean and variance values are then fed to the squashing function to classify the occupancy of the query locations (Figure 1(d) and Figure 2(d)).

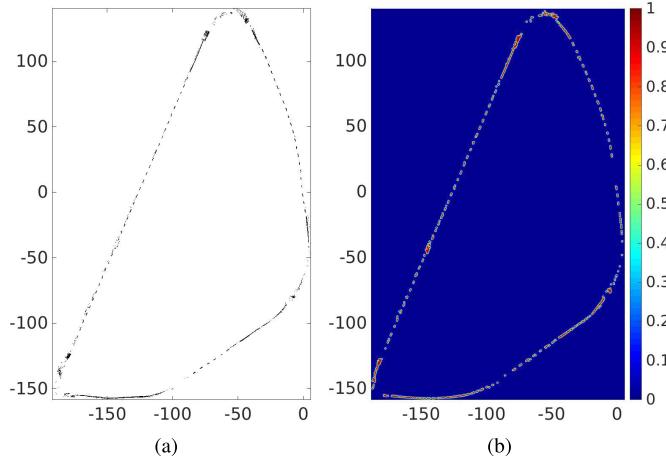


Fig. 4. Generated maps from the road marking data of Track 1. (a) binary OGM and (b) likelihood filed map of the OGM.

In order to validate the resulting GPOM, OGMs configured with 0.10 m, 0.20 m and 0.30 m resolution were built using the same curb data. These resolutions depict fine to moderate grained grids that are commonly employed in localization problems [7]. The 0.10 m resolution OGM of Track 1 and Track 2 are presented in Figure 1(c) and Figure 2(c).

Figure 3 presents five amplified parts (denoted by letters A to E) of GPOM and OGM built with different resolutions. It is noticed the presence of gaps in the OGM independently of the resolution, while GPOM generates a uniform surface. From the overall view of Figure 3, the road area (blue color) and the curb obstacle (red color) is better highlighted in GPOM than in OGM. Moreover, GPOM defines a continuous region of transition between occupied and free areas.

2) Road Marking Mapping: Differently from curbs, road markings were represented exclusively through OGM because of the irregularity (in contrast to curbs, road markings are not present in every Velodyne frame) and they are used as an auxiliary support during localization. In this sense, road markings are mainly designated to provide supplementary information for measurement likelihood model at places that curbs fail to obtain accurate position estimation.

The road marking mapping process consists on projecting the georeferenced road marking point cloud in a discrete binary OGM. As a result, rather than using occupancy grid mapping algorithm, cells associated to road markings are directly marked as occupied. In this sense, the final map is a binary grid where the cells inform the presence or absence of road marking.

The obtained road marking maps for Track 1 and Tack 2 are shown in Figure 4 and Figure 5. Maps were configured with 0.10 m cell resolution.

B. Vehicle Localization in Urban Environments

The vehicle localization is performed by MCL method adapted to support curb (GPOM) and road marking (OGM) maps. The position prior is obtained by a high-precision GNSS device. In this sense, the MCL follows tracking localization approach. The odometry motion model relies on wheel encoder measurements. In all experiments, MCL was configured to use

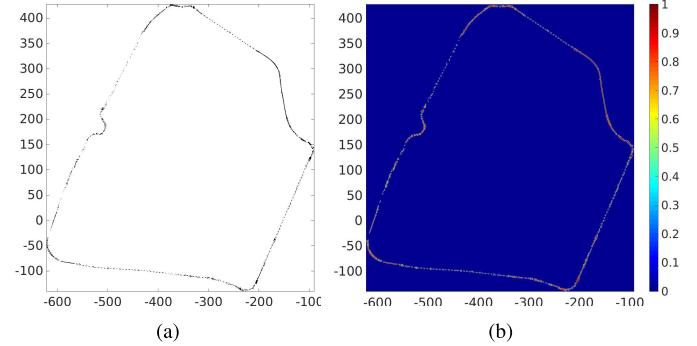


Fig. 5. Generated maps from the road marking data of Track 2. (a) binary OGM and (b) likelihood filed map of the OGM.

1000 particles. Here MCL combined with OGM (standard MCL) and MCL combined with GPOM is referred as MCL-GPOM and MCL-GPOM, respectively. Basically, MCL-OGM solution is employed to evaluate MCL-GPOM results. The same datasets were used for MCL-OGM and MCL-GPOM experiments.

For the MCL-GPOM, occupancy values of the scenario were precomputed in order to make the method computationally tractable. For this, occupancy of a set of evenly spaced points (0.10 m spacing) was evaluated by the GPOM model. The inference results were stored in a matrix and accessed during likelihood calculation. Despite approximating the GPOM to a fine-grained grid, it still provides richer information of the environment than OGM due to the estimation of occupancy of unseen areas. Particle weights are updated using the multivariate normal probability distribution function as in Equation 25.

We conducted localization experiments in scenarios of Tracks 1 and 2 during low traffic. The vehicle run at approximately 30 km/h to 45 km/h. In each track, we performed one experiment using only curb measurement and a second experiment using both curb and road marking measurements. To incorporate road marking observations in the curb measurement model, multimap likelihood model is employed (Equation 27). We highlight that the datasets used in the localization experiment are different from the mapping datasets.

The localization results are given by longitudinal (x), lateral (y), orientation (θ) and euclidean (xy) position offsets. These errors (offsets) are given by the absolute difference of the estimated and the ground truth position (gathered by a GNSS-RTK device). The following subsections describe the localization results obtained in each scenario.

1) Track 1: Localization errors (mean and variance) of the methods using just curb and using both curb and road marking observations are listed in Table I(a) and Table I(b), respectively.

Table I(a) shows that MCL-GPOM delivered lower longitudinal, lateral and euclidean errors than MCL-OGM, despite a slightly higher orientation error. The proposed solution delivered lateral and longitudinal errors more than 0.10 m smaller compared to the best MCL-OGM result (0.30 m grid). Analyzing the euclidean error which takes into account

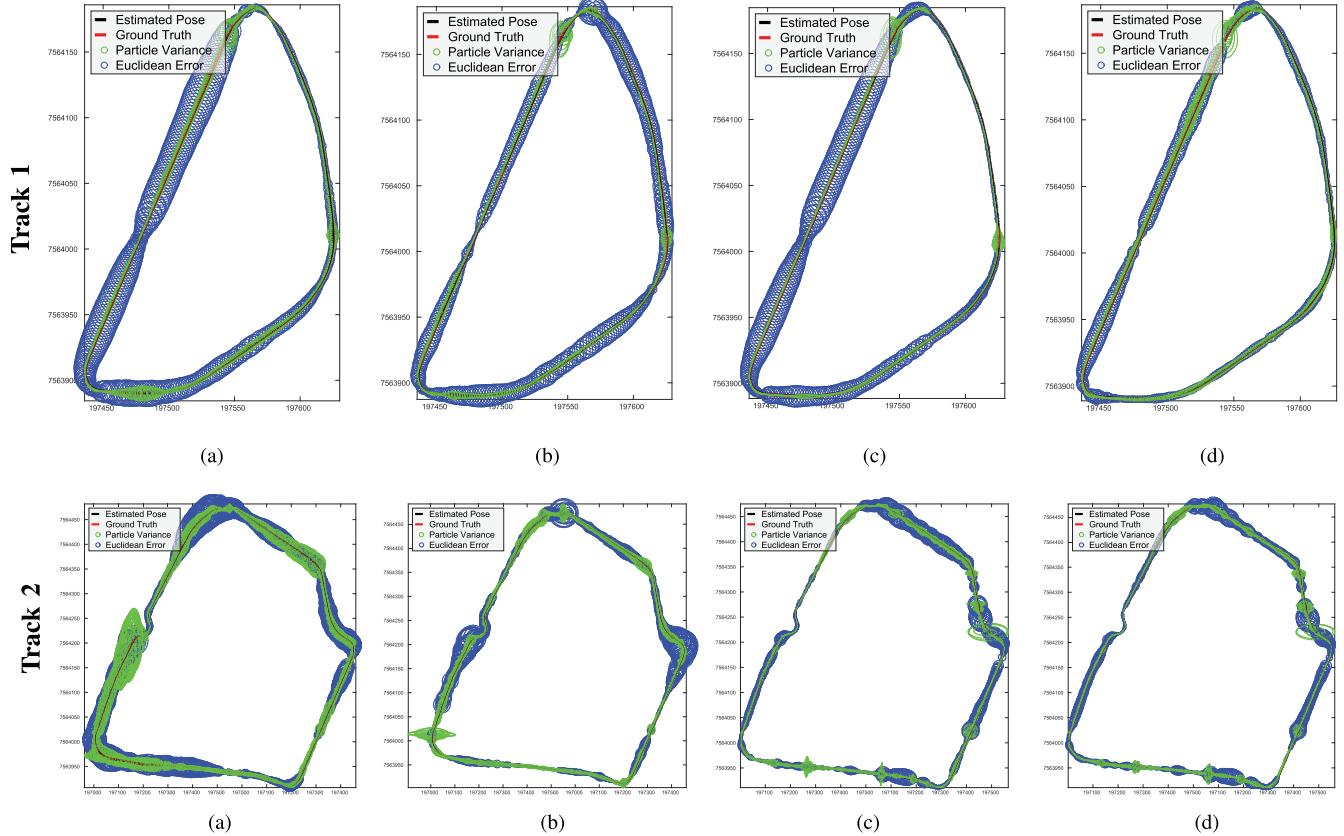


Fig. 6. Localization errors superimposed in the traveled path of Track 1 and Track 2. (a)–(b) MCL-OGM using curb and both curb and road marking observations. (c)–(d) MCL-GPOM using curb and both curb and road marking observations. Black and red lines depicts the estimated and ground truth poses, respectively. Green and blue circles represent the particle variances and euclidean errors, respectively.

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

Dataset	Length	Map Size	Resolution (OGM)
Track 1	770 m	199.9 m × 306.9 m	0.10 ~ 0.30 m
Track 2	1749 m	573.0 m × 536.7 m	0.10 ~ 0.30 m

lateral and longitudinal components, a 0.15 m lower error was observed. A similar behavior was produced when including the road marking data as shows Table I(b). In this setting, a more significant reduction in euclidean error was observed (approximately 0.25 m).

Error results depicted along the track are illustrated in the first row of Figure 6. The poses estimated by MCL are plotted together with the ground truth. For a better visualization of the displacement between the estimated poses and ground truth, the euclidean error at each location is represented by blue circles. The particle variance (*i.e.* the particle cluster size) which corresponds to the MCL convergence is depicted by green ellipses.

As the curb data is ambiguous in the longitudinal direction, MCL produced larger euclidean errors in a straight path and a smaller error in the curves. Moreover, the error is reduced by adding road marking information in the MCL approaches. This effect is more expressive around longer straight paths. Though

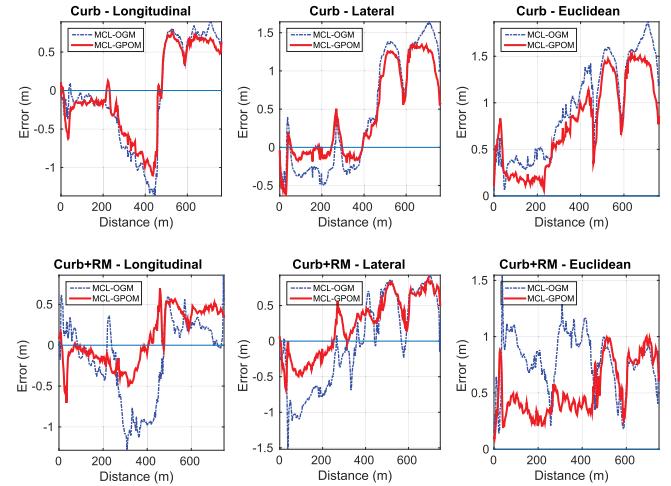


Fig. 7. Localization errors of Track 1. First row: localization using curb observations. Second row: localization using both curb and road marking observations. Values closer to 0 are associated to lower errors. Blue and red lines represent the MCL-OGM and MCL-GPOM approaches, respectively.

the influence of the road marking type in the localization error was not explicitly analyzed, it can be seen in Table II that regions with pedestrian crossing or dashed lane have lower euclidean errors compared to those with continuous lane.

The localization error values (lateral, longitudinal and euclidean) in relation to the traveled distance are shown

TABLE II

TRACK 1 LOCALIZATION ERRORS. ABSOLUTE AVERAGE LONGITUDINAL (x), LATERAL (y), EUCLIDEAN (xy) AND ANGULAR (θ) ERRORS.
 (A) TRACK 1 – MCL USING CURB OBSERVATIONS. (B) TRACK 1 – MCL WITH CURB AND ROAD MARKING OBSERVATIONS

Map	x (m)	y (m)	θ (rad)	xy (m)
OGM 0.10 m	0.6641 ±0.2317	0.5862 ±0.1345	0.0503 ±0.1887	0.9611 ±0.2270
OGM 0.20 m	0.6629 ±0.2479	0.5623 ±0.1116	0.0454 ±0.1574	0.9444 ±0.2232
OGM 0.30 m	0.6449 ±0.2427	0.5360 ±0.1301	0.0442 ±0.1579	0.9096 ±0.2484
GPOM	0.5041 ±0.2192	0.4699 ±0.0797	0.0629 ±0.2503	0.7529 ±0.2069

Map	x (m)	y (m)	θ (rad)	xy (m)
OGM 0.10 m	0.5850 ±0.1177	0.4344 ±0.0973	0.0600 ±0.2508	0.8165 ±0.0792
OGM 0.20 m	0.5587 ±0.1002	0.4133 ±0.0955	0.0625 ±0.2505	0.7889 ±0.0562
OGM 0.30 m	0.5384 ±0.1057	0.4002 ±0.1039	0.0635 ±0.2500	0.7708 ±0.0655
GPOM	0.4059 ±0.0483	0.2795 ±0.0306	0.0560 ±0.1909	0.5194 ±0.0519

TABLE III

TRACK 2 LOCALIZATION ERRORS. ABSOLUTE AVERAGE LONGITUDINAL (x), LATERAL (y), EUCLIDEAN (xy) AND ANGULAR (θ) ERRORS.
 (A) TRACK 2 – MCL USING CURB OBSERVATIONS. (B) TRACK 2 – MCL WITH CURB AND ROAD MARKING OBSERVATIONS

Map	x (m)	y (m)	θ (rad)	xy (m)
OGM 0.10 m	0.6282 ±0.2485	0.6878 ±0.2775	0.0381 ±0.0489	1.0518 ±0.2873
OGM 0.20 m	0.7612 ±0.3972	0.6905 ±0.2913	0.0388 ±0.0335	1.1440 ±0.4360
OGM 0.30 m	1.2023 ±1.7057	1.2200 ±1.0896	0.0489 ±0.0658	1.9489 ±1.9307
GPOM	0.3529 ±0.0873	0.3756 ±0.1297	0.0336 ±0.0170	0.5717 ±0.1556

Map	x (m)	y (m)	θ (rad)	xy (m)
OGM 0.10 m	0.4412 ±0.1756	0.3442 ±0.1028	0.0388 ±0.0496	0.6109 ±0.2183
OGM 0.20 m	0.5067 ±0.4678	0.4434 ±0.3451	0.0422 ±0.0501	0.7255 ±0.7398
OGM 0.30 m	0.8051 ±0.8434	0.7502 ±0.3868	0.0405 ±0.0648	1.2580 ±0.8585
GPOM	0.3618 ±0.0787	0.3575 ±0.0724	0.0335 ±0.0170	0.5550 ±0.1018

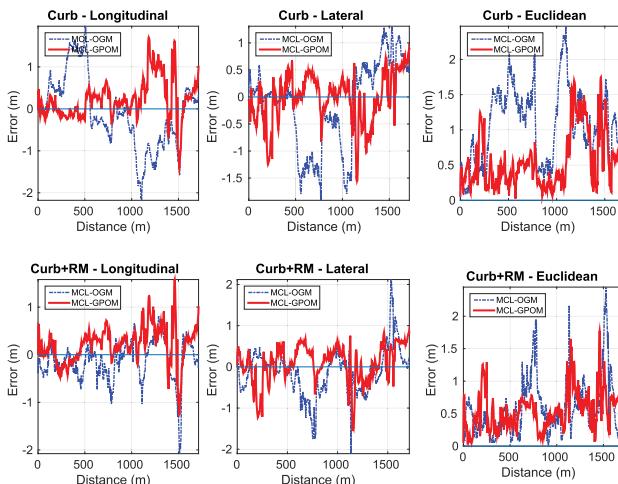


Fig. 8. Localization errors of Track 2. First row: localization using curb observations. Second row: localization using both curb and road marking observations. Values closer to 0 are associated to lower errors.

in Figure 7. Each plot shows both MCL-GPOM (red line) and MCL-OGM (blue line) errors. In these graphs, MCL-GPOM approaches produced errors closer to zero than MCL-OGM. This means that MCL-GPOM returned lower errors than MCL-OGM.

2) *Track 2:* Experiments in Track 2 was performed in the same conditions as in Track 1. Obtained errors of the localization methods using only curb and both curb and road marking observations are listed in Table II(a) and Table II(b),

respectively. This scenario produced slightly higher localization errors for MCL-OGM using just curb features due to the higher sparsity of the map. However, the error was reduced by incorporating road marking measurements. The MCL-GPOM approach delivered lower results than MCL-OGM in all categories. Nonetheless, we didn't observe a considerable error reduction by adding road marking features.

In numerical terms, the GPOM approach resulted in almost 0.30 m lower lateral and longitudinal errors compared to MCL-OGM approach using 0.10 m resolution map (resulted in the best results of MCL-OGM approach) when used just curb observations. This difference is reduced to less than 0.10 m after including the road marking information.

The localization errors of MCL-OGM configured with 0.10 m resolution and MCL-GPOM obtained along the path are illustrated in the second row of Figure 6. It is possible to notice a considerable higher euclidean error and particle variance in the MCL-OGM approaches, especially in the straight paths. As mentioned before, the map sparsity interfered in the measurement likelihood calculation which resulted in larger errors. Detailed graph showing the error value according to the traveled distance is illustrated in Figure 8. Notice that MCL-GPOM errors are closer to zero compared to MCL-OGM.

V. CONCLUSION

This paper proposes a new approach of MCL which uses GPOM (MCL-GPOM) representation to estimate the vehicle position with an accuracy superior to the standard MCL.

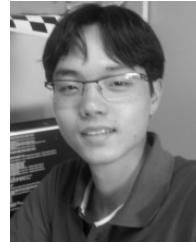
The main advantage of the proposed solution is the possibility to deal with sparse LIDAR data and calculate the measurement likelihood of unobserved areas. The developed localization system relies on a GPOM built from curb data extracted from sparse LIDAR measurements. Thanks to the robustness of the GP inference, occupancy of unobserved areas of the road extent could be estimated and generated a continuous map. As the exclusive use of curb information in the MCL may produce ambiguous particle positions, the road marking data was additionally employed.

Precisely, the likelihood function is based on multivariate normal probability density function that takes into account the occupancy and variance values of whole LIDAR beam. We performed localization experiments in two scenarios and the results compared to the MCL using OGM (MCL-OGM). In the first one, it was obtained at least 0.13 m lower longitudinal and lateral errors for MCL-GPOM. The euclidean error was 0.16 m smaller for MCL-GPOM. Including the road marking observations, our solution still provided better results and reducing the longitudinal, lateral and euclidean errors to 0.4059 m, 0.2795 m and 0.5194 m respectively.

For future works, we propose to represent the road marking data through GPOM and integrate with the GPOM built with curb data. Moreover, the computational complexity will be explored to reduce the processing time of the localization method. This also includes the GP inference computation in each MCL timestamp instead of using precomputed values of mean and variance of the environment space. Additionally, information gathered from other sensors (*e.g.* camera, radar) is pretended to be fused in the GPOM map for a richer representation of the urban environment.

REFERENCES

- [1] L. C. Fernandes *et al.*, “CaRINA intelligent robotic car: Architectural design and applications,” *J. Syst. Archit.*, vol. 60, no. 4, pp. 372–392, 2014.
- [2] T. Luettel, M. Himmelsbach, and H.-J. Wuensche, “Autonomous ground vehicles—Concepts and a path to the future,” *Proc. IEEE*, vol. 100, pp. 1831–1839, May 2012.
- [3] 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 *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2012, pp. 2640–2646.
- [4] L. Guo, M. Yang, B. Wang, and C. Wang, “Occupancy grid based urban localization using weighted point cloud,” in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 60–65.
- [5] S. T. O’Callaghan and F. T. Ramos, “Gaussian process occupancy maps,” *Int. J. Robot. Res.*, vol. 31, no. 1, pp. 42–62, 2012.
- [6] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning* (Adaptive Computation And Machine Learning). Cambridge, MA, USA: MIT Press, 2005.
- [7] J. Levinson *et al.*, “Towards fully autonomous driving: Systems and algorithms,” in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2011, pp. 163–168.
- [8] H. Deusch, J. Wiest, S. Reuter, D. Nuss, M. Fritzsche, and K. Dietmayer, “Multi-sensor self-localization based on maximally stable extremal regions,” in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2014, pp. 555–560.
- [9] J. Rohde, B. Völz, H. Mielenz, and J. M. Zöllner, “Precise vehicle localization in dense urban environments,” in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 853–858.
- [10] P. Oguz-Ekim, K. Ali, Z. Madadi, F. Qutin, and W. P. Tay, “Proof of concept study using DSRC, IMU and map fusion for vehicle localization in GNSS-denied environments,” in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 841–846.
- [11] K. Yoneda, H. Tehrani, T. Ogawa, N. Hukuyama, and S. Mita, “Lidar scan feature for localization with highly precise 3-D map,” in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2014, pp. 1345–1350.
- [12] J. K. Suhr, J. Jang, D. Min, and H. G. Jung, “Sensor fusion-based low-cost vehicle localization system for complex urban environments,” *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1078–1086, May 2017.
- [13] S. Bauer, Y. Alkhorshid, and G. Wanielik, “Using high-definition maps for precise urban vehicle localization,” in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 492–497.
- [14] B. Ferris, D. Fox, and N. Lawrence, “WiFi-SLAM using Gaussian process latent variable models,” in *Proc. 20th Int. Joint Conf. Artif. Intell.*, San Francisco, CA, USA, 2007, pp. 2480–2485.
- [15] C. Plagemann, K. Kersting, P. Pfaff, and W. Burgard, “Gaussian beam processes: A nonparametric Bayesian measurement model for range finders,” *Robot. Sci. Syst.*, 2007.
- [16] A. Y. Hata and D. F. Wolf, “Feature detection for vehicle localization in urban environments using a multilayer LIDAR,” *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 2, pp. 420–429, Feb. 2016.
- [17] M. Seeger, “Gaussian processes for machine learning,” *Int. J. Neural Syst.*, vol. 14, no. 2, pp. 69–106, Apr. 2004.
- [18] C. Stachniss and H. Kretzschmar, *Pose Graph Compression for Laser-Based SLAM*. Cham, Switzerland: Springer, 2017, pp. 271–287.
- [19] S. Kim and J. Kim, “Building occupancy maps with a mixture of Gaussian processes,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2012, pp. 4756–4761.



Alberto Y. Hata received the Ph.D. degree from the University of São Paulo (ICMC/USP) in 2016. He is currently a member of the Mobile Robotics Laboratory, ICMC/USP, and a Post-Doctoral Researcher in the field of machine learning applied for vehicle perception. He has been involved in mobile robotics, machine learning, and image processing. His current research interests are autonomous car perception, Bayesian learning, and robot localization.



Fabio T. Ramos received the B.Sc. and M.Sc. degrees in mechatronics engineering from the University of São Paulo, Brazil, in 2001 and 2003, respectively, and the Ph.D. degree from The University of Sydney, Australia, in 2007. From 2007 to 2010, he was an Australian Research Council Research Fellow with the Australian Centre for Field Robotics and an ARC DECRA Fellow from 2012 to 2014. He is currently an Associate Professor in machine learning and robotics with the School of Information Technologies, The University of Sydney. His research includes statistical machine learning techniques for large-scale data fusion with applications in robotics, mining, environmental monitoring, and healthcare.



Denis F. Wolf received the Ph.D. degree in computer science from the University of Southern California in 2006. He is currently the Director of the Mobile Robotics Laboratory, University of São Paulo (ICMC/USP), and a member of the Center for Robotics, USP. He is also an Associate Professor with the Department of Computer Systems, ICMC/USP. His main research interest is the development of robotic systems for field operation and intelligent transportation systems. His current research interests include computer vision, sensor fusion, and machine learning applied to terrestrial and aerial autonomous robots and vehicles.