

# Curb Detection for a Pedestrian Robot in Urban Environments

Jérôme Maye, Ralf Kaestner, and Roland Siegwart

Autonomous Systems Lab, ETH Zurich, Switzerland

email: {jerome.maye, ralf.kaestner, roland.siegwart}@mavt.ethz.ch

**Abstract**—In this paper, we address the problem of curb detection for a pedestrian robot navigating in urban environments. We devise an unsupervised method that is mostly view-independent, makes no assumptions about the environment, restricts the set of hand-tuned parameters, and builds on sound probabilistic reasoning from the input data to the outcome of the algorithm. In our approach, we construct a piecewise planar model of the environment and determine curbs at plane segment boundaries. Initially, we sense the environment with a nodding laser range-finder and project the 3D measurements into an efficient Digital Elevation Map (DEM). Each cell of the DEM maintains an error model that is propagated throughout the entire algorithm. Plane segments are further estimated with a mixture of linear regression models on the DEM. Here, we propose an original formulation of the standard Expectation-Maximization (EM) algorithm for mixture models. Specifically, in the E-step, responsibilities are computed with a Conditional Random Field (CRF) that introduces dependencies between the covariates of the mixture model. A graph-based segmentation of the DEM provides an estimate of the number of planes and initial parameters for the EM. We show promising results of the algorithm on simulated and real-world data.

## I. INTRODUCTION

Urban areas are highly complex environments which introduce numerous challenges to autonomous service robots. In particular, for a safe and reliable navigation, a robot should be able to accurately detect curbs. Curbs usually appear at the borders between streets and sidewalks. The knowledge of curb positions and characteristics can beneficially enhance metric maps with traversability information relevant to navigation. For instance, depending on its physical capabilities, a robotic platform could only drive harmlessly over curbs of a given height when crossing a street.

Amongst the difficulties related to this task, curbs might exhibit various curvatures and heights, and be perceived from different viewpoints. In contrast to autonomous cars, pedestrian robots can indeed make few assumptions about the structure of the environment. Furthermore, the sensing device noise model should be introduced to distinguish between real curbs and measurement noise. Ideally, the algorithm should run on-line and in real-time.

In this paper, we devise an unsupervised method to curb detection that covers most of the aforementioned requirements. Our approach attempts to construct a piecewise planar model of the environment and determines curbs at plane segment boundaries. Initially, we sense the environment with a nodding laser range-finder and project the 3D measurements into an efficient Digital Elevation Map (DEM). Each cell of the DEM maintains an error model that is

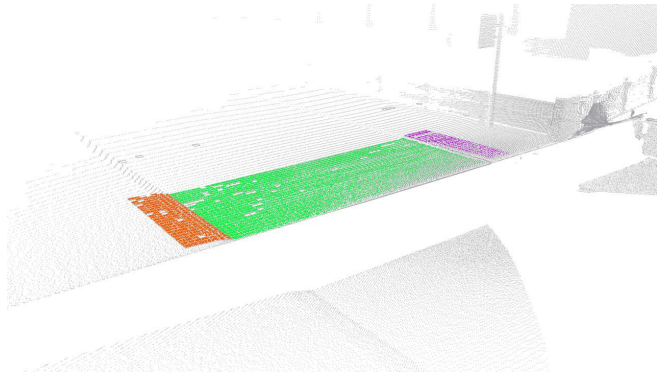


Fig. 1. Exemplary output of our curb detection algorithm (best viewed in color). Plane segments are reconstructed from a DEM patch and curbs flagged at their boundaries. Colors encode cells assignments to the planes.

propagated throughout the entire algorithm. Plane segments are further estimated with a mixture of linear regression models. Here, we propose an original formulation of the standard Expectation-Maximization (EM) algorithm for mixture models. Specifically, in the E-step, the responsibilities are computed with a Conditional Random Field (CRF) that introduces dependencies between the covariates of the mixture model. A graph-based segmentation of the DEM provides an estimate of the number of planes and initial parameters for the EM. Sequential and fast estimation of DEM patches in the surrounding of the sensor while the robot drives ensure on-line operation. Fig. 1 shows a typical output of our algorithm.

Clearly, the main contribution of the paper is its strict probabilistic interpretation from the sensing process to the final plane estimation and segmentation. We are thus able to elegantly incorporate and reflect the induced measurement noise. Moreover, our method is view-independent and requires no particular prior knowledge about the environment. The set of free parameters is solely related to the sensor characteristics and involves no hand-tuning. Finally, a direct implementation of the algorithm from the paper should be straightforward.

The remainder of the paper is structured as follows. Section II summarizes the previous works related to ours. Section III introduces our statistical models and derives the related inference methods. Section IV is concerned with implementation and algorithmic details. Section V demonstrates the validity of the method through extensive qualitative and quantitative analysis. Section VI outlines our conclusions and provides some insights for future work.

## II. RELATED WORK

The problem of curb or *step* detection has mainly been studied in the context of Intelligent Transportation Systems (ITS), covering an extensive set of sensing modalities and algorithms. In ITS, one might assume a typical experimental setting where a car drives on a street and curbs are situated on the left and right side of the vehicle. Therefore, most of these approaches are inappropriate as such for a pedestrian robot navigating in cities. In this situation, curbs will indeed appear under multiple viewpoints. We hence review the main influential contributions to the field and relate them to our method.

In [1], Oniga *et al.* employ a dense stereo-vision system to capture a 3D point cloud, which is further transformed into a Digital Elevation Map (DEM). Curbs are represented as third-order polynomials. Candidate curb points are extracted with a Canny edge detector. A RANdom SAMple Consensus (RANSAC) polynomial fitting is then applied to perform outlier rejection and find the polynomial coefficients. The location of the curbs and their heights are finally obtained with some further refinement steps. In comparison to this method, we use the same measurement representation (DEM). However, we draw a clear and sound probabilistic model from the sensing device to the curb detection and limit the number of hand-tuned parameters.

Closer to our approach, Siegemund *et al.* [2] proposed a promising method that extracts curbs from dense stereo-vision data. We actually take inspiration from their ideas and solve their major drawbacks. In this paper, as mentioned above, they assume a strict environment model and they lack a unified probabilistic model. Moreover, their curb models can only represent a limited set of curbs. For instance, they cannot model T junctions or roundabouts.

In [3], Shin *et al.* use a similar setup as ours, i.e., a mobile robot equipped with a laser range-finder. They however stick to a restricted environment model and their tilted laser only provides a single laser line. Their algorithm is mostly engineered to fit their particular application and setup, and again does not build on sound probabilistic model.

An alternative application of step detection is presented in [4]. In this paper, Pradeep *et al.* aim at mobility aids for visually impaired people, as a complement to the white cane or the dog. This naturally imposes restrictions in the sensing device, in this case a wearable and cheap stereo camera. Their curb detection algorithm is based on the same motivation as ours, i.e., building a piecewise planar model of the scene. In their implementation, point-wise normal vectors are firstly estimated with Principal Component Analysis on a local neighborhood and RANSAC for outlier rejection. Tensor voting is then applied for finding globally consistent plane normals and a final clustering step extracts the plane segments. Although it solves many of the above issues, this method might also suffer from the lack of any underlying probabilistic models.

In [5], Yuan and Manduchi also developed an algorithm for visually impaired people, working with a custom sensing

device called a "Virtual White Cane". Their method uses a Jump-Markov Process to detect geometric singularities in the range measurements. Despite its statistical foundations, it will not satisfy our application requirements and provide a proper plane estimation.

In comparison to other works on segmentation of 3D range data such as [6], our method is fully unsupervised and thus does not require any training data. Furthermore, the dimensionality of our model adapts to the data and, in addition to a labeling of 3D point clouds, we estimate a full generative model.

## III. MODEL

In this section, starting from the measurement process, we establish a statistical model for curb detection and derive appropriate inference algorithms.

### A. Measurement Representation

A nodding laser range-finder produces scan measurements  $\mathbf{s}_i = [r_i, \theta_i, \psi_i]^T$  that are transformed into their corresponding Cartesian 3D coordinates  $\mathbf{p}_i = [x_i, y_i, z_i]^T$ , where  $r_i$  is a range measurement,  $\theta_i$  a pitch angle, and  $\psi_i$  a bearing angle. The sensing device has an error model which is typically a function of  $\mathbf{s}_i$ . From a complete laser sweep, we obtain a point cloud representation  $\mathcal{P} = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_N\}$ .

Unfortunately, point clouds are inconvenient models in the context of robot navigation, where single points are frequently subject to expensive random access operations, e.g., in the case of collision checks. Therefore, it will be beneficial to project  $\mathcal{P}$  onto a 2D grid  $\{C_1, C_2, \dots, C_M\}$ , with cells  $C_i = \{\mathbf{c}_i, \mathbf{B}_i, h_i\}$ , where  $\mathbf{c}_i = [c_{ix}, c_{iy}]^T$  is the cell center,  $\mathbf{B}_i = [\mathbf{ul}_i, \mathbf{lr}_i]^T$  its bounding box with upper left point  $\mathbf{ul}_i$  and lower right point  $\mathbf{lr}_i$ ,  $h_i$  its height.

To account for the noise induced by the aforementioned measurement and discretization process, we model the height  $h_i$  as a normal distribution  $p(h_i | \mu_{h_i}, \sigma_{h_i}^2) = \mathcal{N}(h_i | \mu_{h_i}, \sigma_{h_i}^2)$ , with mean  $\mu_{h_i}$  and variance  $\sigma_{h_i}^2$  to be estimated. To this end, we collect the set of points  $\mathcal{I}_i = \{z_j | \mathbf{p}_j \in \mathcal{P}, [x_j, y_j]^T \subset \mathbf{B}_i\}$  and adopt a purely Bayesian approach [7]. This involves setting a joint prior  $p(\mu_{h_i}, \sigma_{h_i}^2)$  over the parameters, computing the posterior  $p(\mu_{h_i}, \sigma_{h_i}^2 | \mathcal{I}_i)$ , and ultimately formulating a posterior predictive density  $p(\hat{h}_i | \mathcal{I}_i)$  that will be subsequently termed our *cell model*. We choose a conjugate normal inverse-gamma prior that incorporates the sensor and discretization noise in its hyperparameters. The resulting posterior predictive density is thus a Student's *t*-distribution. We denote the mean and variance of this density  $\hat{\mu}_{h_i}$  and  $\hat{\sigma}_{h_i}^2$  respectively. Apart from its elegance, this Bayesian method can plainly reflect our noise model and remains robust to outliers, especially in case of few measurements.

In the literature, the 2D grid employed in this paper is commonly designated a *Digital Elevation Map* (DEM). We will therefore adopt the notion and henceforth use it to refer to this kind of grid. The choice of a DEM representation is mainly guided by the application scope of the algorithm. It may for example be interpreted as a traversability map for the planning process. It is furthermore convenient for defining

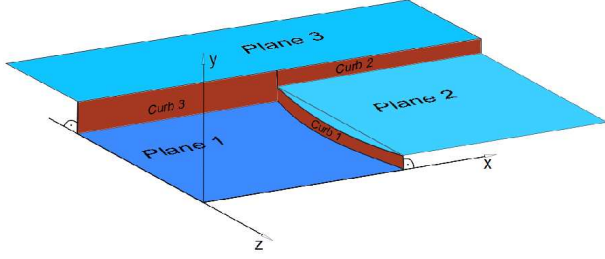


Fig. 2. The proposed environment model is composed of a set of plane segments. Curbs are defined as the boundaries between plane segments.

Regions of Interest (ROI) in  $\mathcal{P}$  and for facilitating further computations. Finally, it provides constant-time lookup to the cells, either addressed by their geometrical coordinates or row-column indices. We shall discuss later on the issue of cell discretization.

### B. Environment Model and Inference Task

We assume a piecewise planar environment, i.e., the observed scene is composed of a set of plane segments. Boundaries between plane segments define local height discontinuities that we shall henceforth term *curbs*. Fig. 2 depicts a typical environment model. We shall remind here that our algorithm is not restricted to this typical configuration. Indeed, it can properly cope with any plane inclinations or lowered curbs for wheelchairs or bikes.

On the one hand, the major inference task therefore boils down to discovering those plane segments, and on the other hand, to determining the assignment of the DEM cells to these latter. To this end, we model the environment with a conditional mixture model, namely a *mixture of linear regression* model. According to the discussions in [8], we may thus state the following generative process for the predictive mean height value:

$$p(\hat{\mu}_{h_i} | \mathbf{c}_i, \Theta) = \sum_{k=1}^K \pi_k \mathcal{N}(\hat{\mu}_{h_i} | \mathbf{w}_k^T \boldsymbol{\phi}(\mathbf{c}_i), \sigma_k^2), \quad (1)$$

where  $\Theta = \{\boldsymbol{\pi}, \mathbf{W}, \boldsymbol{\sigma}^2\}$  is the set of adaptive parameters to be estimated,  $\boldsymbol{\pi} = \{\pi_k\}$  are the mixture weights,  $\mathbf{W} = \{\mathbf{w}_k\}$  the regression coefficients,  $\boldsymbol{\sigma}^2 = \{\sigma_k^2\}$  the regression variances, and  $\boldsymbol{\phi}(\mathbf{c}_i) = [1 \ \mathbf{c}_i]^T$  the explanatory variables or *covariates*. The cell variance  $\hat{\sigma}_{h_i}^2$  can be introduced in the parameter estimation as we shall demonstrate below.

For solving the classification task, we introduce an additional categorical latent variable  $\mathbf{I}_i$ , which has a 1-of-K representation such that its prior distribution is defined as  $p(I_{ik} = 1) = \pi_k$ . Intuitively, the posterior distribution  $p(I_{ik} = 1 | \hat{\mu}_{h_i})$  or *responsibility*  $\gamma_{ik}$  represents the probability for the cell  $C_i$  to belong to plane  $k$ .

### C. Parameter Estimation

Given an instance of a DEM, the parameter set  $\Theta$  of the mixture model has to be estimated. We adopt here a Maximum-Likelihood (ML) approach, i.e., we search for the

parameters  $\hat{\Theta}$  that maximize the likelihood function or "best" explain the data.

We define the vector of all mean height values as  $\hat{\boldsymbol{\mu}}_h = [\hat{\mu}_{h_1} \dots \hat{\mu}_{h_M}]^T$ , the matrix of covariates as  $\boldsymbol{\Phi} = [\boldsymbol{\phi}(\mathbf{c}_1) \dots \boldsymbol{\phi}(\mathbf{c}_M)]^T$ , and the matrix of latent variables as  $\mathbf{L} = [\mathbf{I}_1 \dots \mathbf{I}_M]^T$ . Due to singularities, there is no analytical solution to the direct maximization of the likelihood function  $p(\hat{\boldsymbol{\mu}}_h | \Theta, \boldsymbol{\Phi})$ . An Expectation-Maximization (EM) algorithm [9] is the prevalent choice in the literature for solving this kind of problem. If, for each point, we were given its class assignment, i.e., we would observe a *complete* data set, the maximization of the complete-data likelihood  $p(\hat{\boldsymbol{\mu}}_h, \mathbf{L} | \Theta, \boldsymbol{\Phi})$  would be simplified. Although latent variables are not observed, we can compute their posterior distribution  $p(\mathbf{L} | \hat{\Theta}^{(t)}, \boldsymbol{\Phi})$  using an initial parameter estimate  $\hat{\Theta}^{(t)}$ . In the E-step of the EM algorithm, the expectation of the complete-data likelihood under this posterior is firstly calculated. In the M-step, a new parameter set  $\hat{\Theta}^{(t+1)}$  is obtained by maximization of this likelihood. The algorithm iterates between these steps until convergence of the likelihood.

More specifically, in a standard EM for mixture of linear regression model, we compute responsibilities  $\gamma_{ik}$ , or posterior class assignments, using Bayes rule in the E-step with

$$p(I_{ik} = 1 | \hat{\mu}_{h_i}, \mathbf{c}_i, \hat{\Theta}^{(t)}) = \gamma_{ik} \propto \pi_k \mathcal{N}(\hat{\mu}_{h_i} | \mathbf{w}_k^T \boldsymbol{\phi}(\mathbf{c}_i), \sigma_k^2). \quad (2)$$

Here, the attentive reader shall have noticed that we have used conditional independence of the  $\mathbf{I}_i$  given the data, that is:

$$p(\mathbf{L} | \hat{\boldsymbol{\mu}}_h, \boldsymbol{\Phi}, \hat{\Theta}^{(t)}) = \prod_{i=1}^M p(\mathbf{I}_i | \hat{\mu}_{h_i}, \mathbf{c}_i, \hat{\Theta}^{(t)}). \quad (3)$$

In the M-step of the algorithm, the maximization yields the following parameter estimates  $\hat{\Theta}^{(t+1)}$ :

$$\mathbf{w}_k = (\boldsymbol{\Phi}^T \mathbf{R}_k \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T \mathbf{R}_k \hat{\boldsymbol{\mu}}_h, \quad (4)$$

$$N_k = \sum_{i=1}^M \gamma_{ik}, \quad (5)$$

$$\sigma_k^2 = \frac{1}{N_k} (\hat{\boldsymbol{\mu}}_h - \boldsymbol{\Phi} \mathbf{w}_k)^T (\hat{\boldsymbol{\mu}}_h - \boldsymbol{\Phi} \mathbf{w}_k), \quad (6)$$

$$\pi_k = \frac{N_k}{M}, \quad (7)$$

where  $\mathbf{R}_k = \text{diag}(\gamma_{ik})$  contains the responsibilities for component  $k$  in a diagonal matrix.

### D. Belief Propagation for Posterior Class Assignments

As we have pointed out in (3), the computation of the joint posterior class assignment  $p(\mathbf{L} | \hat{\boldsymbol{\mu}}_h, \boldsymbol{\Phi}, \hat{\Theta}^{(t)})$  can be handled independently for each  $\mathbf{I}_i$  in the standard EM algorithm. The sought marginals  $\gamma_{ik}$  are therefore straightforwardly calculated. However, the nature of our problem suggests that we should introduce dependencies between neighboring DEM cells. Indeed, geometrically close points are more

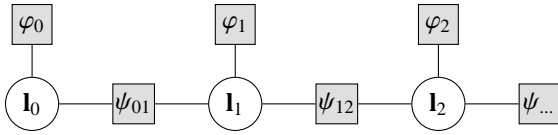


Fig. 3. Excerpt of the factor graph used for Belief Propagation (BP). Latent variables ( $\mathbf{I}_i$ ) are displayed with circles and factor nodes ( $\psi_{ij}, \varphi_i$ ) with squares. The factor graph expresses our specific joint factorization.

likely to lie in the same plane. We hence define an undirected graph  $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ , with vertices  $v_i \in \mathcal{V}$  and edges  $(v_i, v_j) \in \mathcal{E}$  connecting neighboring vertices. Each DEM cell  $C_i$  is assigned to a vertex  $v_i$  having a 4-connected neighborhood.

Borrowing in the literature of graphical models, we can express the joint posterior class assignment factorization with a Conditional Random Field (CRF) [10] on the graph  $\mathcal{G}$  as:

$$p(\mathbf{L} | \hat{\boldsymbol{\mu}}_h, \boldsymbol{\Phi}, \hat{\Theta}^{(t)}) \propto \prod_{v_i \in \mathcal{V}} \varphi_i(\hat{\boldsymbol{\mu}}_{h_i}, \mathbf{I}_i, \hat{\Theta}^{(t)}, \boldsymbol{\Phi}) \prod_{(v_i, v_j) \in \mathcal{E}} \psi_{ij}(\hat{\boldsymbol{\mu}}_{h_i}, \hat{\boldsymbol{\mu}}_{h_j}, \mathbf{I}_i, \mathbf{I}_j), \quad (8)$$

where  $\varphi_i(\cdot)$  and  $\psi_{ij}(\cdot)$  are the *node potentials* and *edge potentials* respectively. The potential functions are positively defined functions and the normalization is ensured by the *partition function*  $\mathbf{Z}(\boldsymbol{\Phi})$ . Intuitively, a node potential reflects the likelihood of  $\hat{\boldsymbol{\mu}}_{h_i}$  being labeled  $\mathbf{I}_i$ , and an edge potential the joint likelihood of  $\hat{\boldsymbol{\mu}}_{h_i}$  and  $\hat{\boldsymbol{\mu}}_{h_j}$  being labeled  $\mathbf{I}_i$  and  $\mathbf{I}_j$ .

Given the formulation in (8), two inference tasks are of interest. On the one hand, one can derive the marginals  $p(\mathbf{I}_i | \hat{\boldsymbol{\mu}}_{h_i}, \mathbf{c}_i, \hat{\Theta}^{(t)})$ , and on the other hand, one can find the Maximum A Posteriori (MAP) joint state  $\mathbf{L}^{\text{MAP}}$  of the CRF. The former question is addressed with the sum-product algorithm, while the latter with the max-product, which give exact results for tree. These are two instances of Belief Propagation (BP). Even though our graph  $\mathcal{G}$  contains loops, BP can yield approximate results [11] and is in this case called *loopy BP*. Fig. 3 shows an excerpt of the factor graph [12] on which the inference operates.

We express the node potentials as follows:

$$\varphi_i(\hat{\boldsymbol{\mu}}_{h_i}, \mathbf{I}_i, \hat{\Theta}^{(t)}, \boldsymbol{\Phi}) = \pi_k \mathcal{N}(\hat{\boldsymbol{\mu}}_{h_i} | \mathbf{w}_k^T \boldsymbol{\phi}(\mathbf{c}_i), \sigma_k^2). \quad (9)$$

We note here that the CRF does not require a normalized potential. Apart from the normalizer, this expression is similar to (2). As expected, BP will hence converge to the same marginals  $\gamma_{ik}$  if edge potentials are omitted.

In analogy to the approach presented in [2], we express the inter-node dependencies by means of two symmetric sigmoid functions. Therefore, we define the edge potentials as

$$\psi_{ij}(\hat{\boldsymbol{\mu}}_{h_i}, \hat{\boldsymbol{\mu}}_{h_j}, \mathbf{I}_i, \mathbf{I}_j) = \begin{cases} 1 - (1 + \exp(\sigma_{ij}^2 - d_{ij}))^{-1} & \text{if } \mathbf{I}_i = \mathbf{I}_j \\ (1 + \exp(\sigma_{ij}^2 - d_{ij}))^{-1} & \text{otherwise,} \end{cases} \quad (10)$$

where  $d_{ij} = |\hat{\boldsymbol{\mu}}_{h_i} - \hat{\boldsymbol{\mu}}_{h_j}|$  is the absolute mean height difference between the neighboring cells  $C_i$  and  $C_j$ . Additionally,  $\sigma_{ij}^2 =$

$\hat{\sigma}_{h_i}^2 + \hat{\sigma}_{h_j}^2$  represents the sum of the involved measurement variances. Consequently, we account for the errors in both cell models  $\hat{\sigma}_{h_i}^2$  and  $\hat{\sigma}_{h_j}^2$ .

#### IV. IMPLEMENTATION DETAILS

We have thus far delivered a formal treatment of our method to curb detection. This section will be dedicated to algorithmic details and specific issues arising with real-world data.

##### A. Model Complexity and Initial Estimates

The statistical model in (1) requires an estimate of the number of mixture components  $K$ . This issue is related to model complexity or model selection and remains a focus of research in itself. As our algorithm should be applicable to any kind of environment, a key prerequisite is to infer  $K$  from the data. To this intent, we opt for an heuristic in the form of a pre-segmentation. Specifically, we adopt the graph-based algorithm from Felzenszwalb and Huttenlocher [13]. Although this method was originally designed for image segmentation, we can adapt it for our purpose, by treating image regions as plane segments.

The algorithm operates on the graph  $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$  defined above and augment it with edge weights  $w((v_i, v_j))$  proportional to the dissimilarity between  $v_i$  and  $v_j$ . The goal of the algorithm is to find a partition of  $\mathcal{V}$  into segments  $\mathcal{S}_i$  that correspond to the connected components of a graph  $\mathcal{G}' = \{\mathcal{V}, \mathcal{E}'\}$ , with  $\mathcal{E}' \subseteq \mathcal{E}$ . We are interested in the specific partition such that vertices in a component have a high similarity and vertices in different components a low similarity. Therefore, edges between vertices in the same component should have a low weight and edges between vertices in different components a high weight. The weight function can be defined by the symmetric Kullback-Leibler divergence between two cells, i.e.,  $w((v_i, v_j)) = D_{KL}(h_i \| h_j) + D_{KL}(h_j \| h_i)$ . We thus take into account the full height distributions, in particular the variances. For normal distributions, the Kullback-Leibler divergence integrates analytically to

$$D_{KL}(h_i \| h_j) = \frac{(\hat{\boldsymbol{\mu}}_{h_i} - \hat{\boldsymbol{\mu}}_{h_j})^2}{2\hat{\sigma}_{h_j}^2} + \frac{1}{2} \left( \frac{\hat{\sigma}_{h_i}^2}{\hat{\sigma}_{h_j}^2} - 1 - \ln \frac{\hat{\sigma}_{h_i}^2}{\hat{\sigma}_{h_j}^2} \right). \quad (11)$$

The algorithm starts with all vertices belonging to a different component. It then iterates over the set of edges ordered by increasing weights. For each edge  $(v_i, v_j) \in \mathcal{E}$  with  $v_i \in \mathcal{S}_k$ ,  $v_j \in \mathcal{S}_l$ , and  $\mathcal{S}_k \neq \mathcal{S}_l$ , the two components are merged if

$$w((v_i, v_j)) \leq MInt(\mathcal{S}_k, \mathcal{S}_l), \quad (12)$$

where  $MInt(\mathcal{S}_k, \mathcal{S}_l) = \min(Int(\mathcal{S}_k) + \tau(\mathcal{S}_k), Int(\mathcal{S}_l) + \tau(\mathcal{S}_l))$ ,  $Int(\mathcal{S}) = \max_{(v_i, v_j) \in \mathcal{S}} w((v_i, v_j))$ , and  $\tau(\mathcal{S}) = s/|\mathcal{S}|$ . Two components should be disconnected if the difference between them is large compared to the internal difference within at least one of the components.  $s$  is a scale parameter that controls the preference for larger components. Vertex components being segmented, an initial estimate  $\hat{\Theta}^{(0)}$  can be

computed with weighted linear regression for starting the EM algorithm.

This heuristic inevitably introduces a free parameter  $s$  that controls the model complexity. Choosing a large  $s$  will potentially result in underestimating  $K$ , while setting it too low in increasing the computational complexity. In our application, we fix it at some empirical value ( $s = 100$ ) that abates the two issues.

### B. Grid Discretization and Missing Data

The grid discretization choice is mainly influenced by the sensor characteristics and sought accuracy. Obviously, a large cell size introduces a proportional estimation error to the curb detection algorithm. On the other hand, depending on the sparsity of the sensor data, a finer size increases the number of cells with no measurement and, at the limit, results in an unconnected graph. Indeed, whenever a cell contains no data, it is flagged as invalid and not considered for the rest of the algorithm. Based on experimental data, we set the cell sizes to  $d_x = 0.1$  [m] and  $d_y = 0.05$  [m], and the grid dimension to  $w = 4$  [m] and  $l = 4$  [m].

### C. Algorithmic Complexity

The algorithmic complexity depends on the aforementioned parameters ( $s, d_x, d_y, w, l$ ). The graph-based segmentation runs in  $O(|\mathcal{E}| \log |\mathcal{E}|)$ , and the standard EM efficiently implemented in  $O(MK)$  with  $M$  the number of valid cells. The final complexity of our method is largely dominated by the addition of the BP pass for each E-step in the EM. This inference method has a run-time complexity in the order of  $O(|\mathcal{E}|K^4)$ . Nevertheless, under practical considerations and adequate parameter setting, we can achieve close to real-time performances as will be shown below.

### D. Implementation

The algorithm has been implemented in a fully template-based C++ library that will be available to the users at the time of publication. The belief propagation inference engine has been borrowed to libDAI [14], an open-source library for discrete approximate inference on graphical models.

## V. EXPERIMENTS

In order to evaluate and validate the approach proposed in this paper, we have conducted experiments on simulated and real-world data. Real-world data has been acquired with a static nodding Laser Range-Finder (LRF) setup. Two different lasers have been mounted, namely a SICK LMS-200 and an Hokuyo UTM-30LX. We also tested our method on a pedestrian robot equipped with a downward-facing SICK LMS-151 LRF that generates 3D point cloud while moving. Simulated data has been generated by sampling from known mixture models and from Morsel, a 3D mobile robot simulator developed in our lab.



Fig. 4. Our pedestrian robot driving in Morsel in the city-like environment used for evaluation.

### A. Experimental Conditions and Quantitative Measures

For the nodding lasers setup, we have recorded 33 3D point clouds with multiple viewpoints from a standard street scene. For the pedestrian robot scenario, data has been generated from a tour in a city center and from a drive in our 3D simulator using a city-like environment (Fig.4).

A quantitative evaluation of our algorithm can be carried out under various interrelated perspectives: curb location in  $x, y$ , curb height in  $z$ , number of planes, assignment of DEM cells to planes, plane parameters, or computation time. Since we do not have ground truth information for the plane parameters, we evaluated the predictive accuracy of a model trained with one point cloud to the others in a similar fashion as [15]. To this end, we collected 19 point clouds from the same position and performed cross-validation. Concretely, we iteratively estimate the parameter set  $\{\hat{\Theta}, \hat{\mathbf{L}}\}$  using one point cloud and evaluate the predictive error on the 18 remaining ones. The quantitative measure is the Root Mean Square (RMS) error of prediction.

To analyze the quality of the segmentation, we manually labeled the point clouds in regions corresponding to plane segments. While accurate ground truth is available from the simulator, real data labeling might suffer from slight subjective errors. Since ground truth and inferred labels can differ in an unsupervised clustering framework, we have used the V-Measure [16] as a quantitative measure. This conditional entropy-based figure ranging from 0 (bad) to 1 (good) combines homogeneity and completeness criteria, and copes with labels mismatches.

In order to further analyze our model, we have sampled point clouds from known mixture of linear regressions and also evaluated in this case the RMS error of the predicted parameters  $\hat{\Theta}$  against their ground truth. In the case of synthetic data, predicted curb location and height, and assignment of cells to planes, can also be quantitatively evaluated. Furthermore, we can judge the robustness and validity of our algorithm on various situations such as T junctions, inclined planes, or lowered curbs.



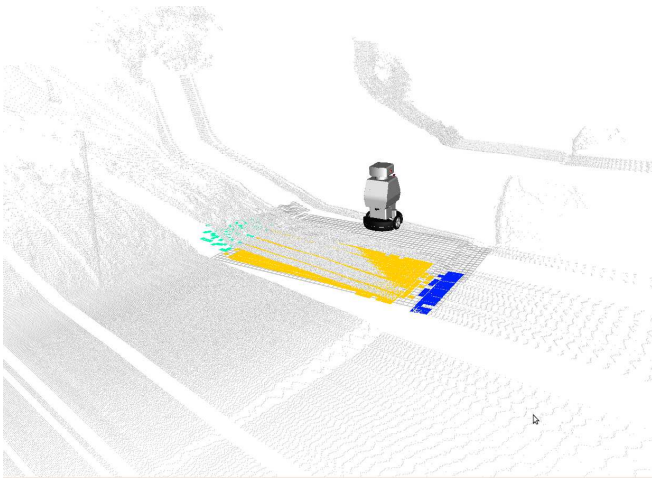


Fig. 5. Example of curb detection from a moving pedestrian robot. Colors represent planes and curbs are located at their boundaries.

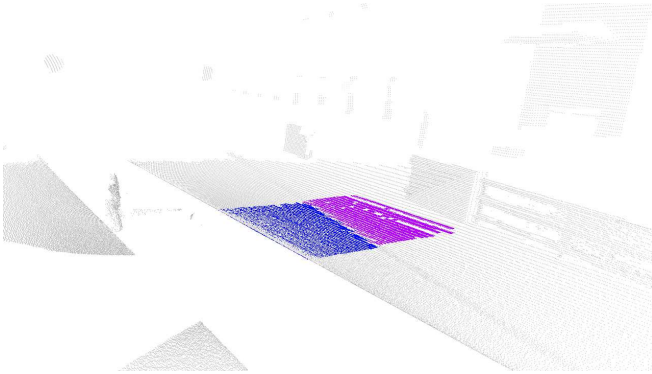


Fig. 6. Example of curb detection in an unfavorable situation. Our algorithm correctly label the planes, and thus curbs, under various viewpoints and experimental settings.

### B. Qualitative Evaluation

Before we proceed with the actual quantitative analysis, we want to give a glimpse on some qualitative results that demonstrate the pertinence of our approach.

In Fig. 5, our pedestrian robot navigates in a city center and labels curbs while driving. Since the point cloud is reconstructed while the robot drives, curbs can only be detected behind the robot in this specific situation. DEM patches are labeled sequentially and we achieve on-line and real-time performance.

Fig. 6 depicts a situation that a pedestrian robot might often encounter when crossing a street. Using sampled data, we demonstrate in Fig. 7 the output of our algorithm in a complex environment containing a T junction, inclined planes, and lowered curbs. These experiments illustrate that our method can cope with multiple viewpoints and environment configurations. Most of the competitive approaches would fail in that cases.

Fig. 8 displays the outcome of the segmentation algorithm on a point cloud, while Fig. 9 shows the results obtained from the standard EM algorithm. As a comparison, our method is applied on the same data and the result is depicted in Fig. 10.



Fig. 7. As exemplified on synthetic data, our method can successfully cope with various environment configurations.

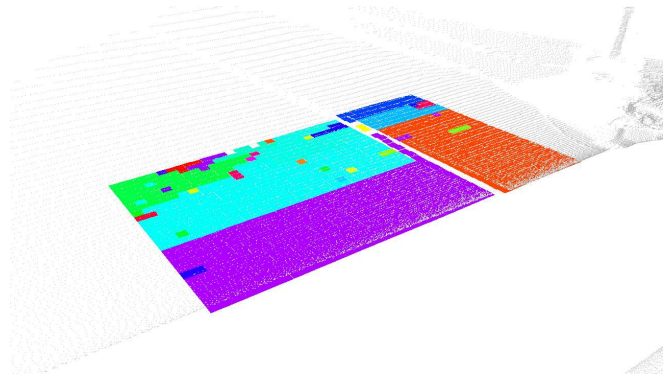


Fig. 8. Graph-based segmentation of the DEM (colors encode assignments of cells to the different components). By keeping the segmentation parameter  $s$  rather low, we ensure no planes are missed, at the cost of over-segmentation.

These experiments clearly highlight the advantages of our method.

### C. Quantitative Evaluation and Discussion

1) *Real-World Data*: Proceeding as mentioned above, i.e., evaluating the prediction accuracy on unseen data and averaging the RMSE over all the datasets, we obtain a RMSE of 0.016 [m] for a DEM containing 1600 cells. This result shows that we are able to accurately estimate the plane parameters and thus reconstruct the scene.

Under the same experimental conditions, the quality of the plane segmentation has been quantitatively evaluated on all the datasets and we obtained a V-Measure of 0.96 on average with a standard deviation of 0.03. We have noticed here that the badly classified cells are mostly due to grid discretization or subjective ground truth.

Our algorithm takes, on a standard dual-core laptop, on average 0.21 [s] with a standard deviation of 0.17. As expected, most of the computation time is spent on the BP-EM part. These timings therefore allow for real-time and on-line operations on a robotic platform.

2) *Synthetic Data*: For the first experiment in our 3D mobile robot simulator, we have evaluated the segmentation quality and obtained a V-Measure of 0.88 on average with

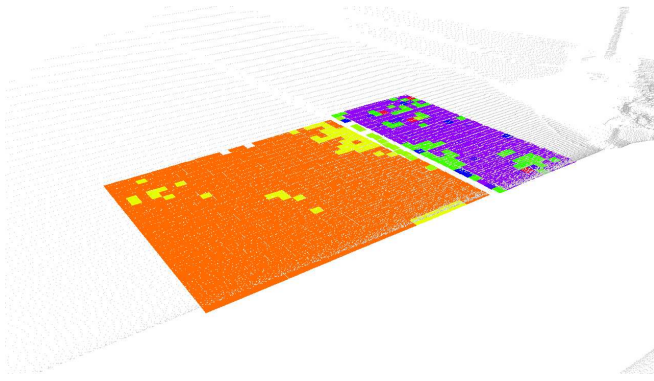


Fig. 9. DEM cell assignments (colors) to plane components using the standard EM algorithm. Due to the initial over-segmentation, the algorithm is not able to fully smooth out the additional planes.

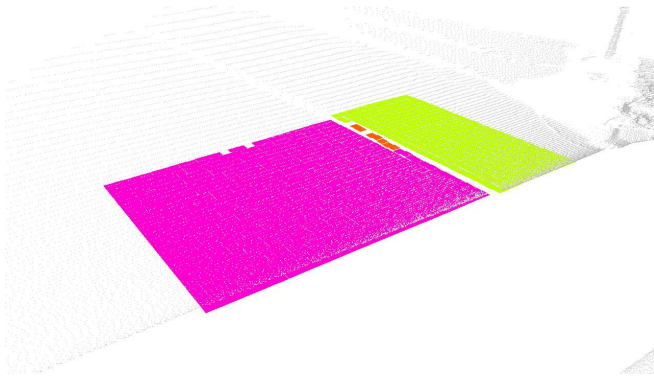


Fig. 10. DEM cell assignments (colors) using our CRF-EM implementation. The MAP state is displayed here. The algorithm is able to smooth out the unnecessary planes and provides a good estimate of the curb positions and heights.

a standard deviation of 0.05. Here again, we have witnessed that the discretization induces the badly classified cells.

In a second experiment, we have generated multiple artificial mixture of linear regression models and sampled 3D point clouds from them. We tried to create situations that were not present in our dataset and varied the different parameters to analyze the robustness of our algorithm. We have evaluated and validated our algorithm under the aforementioned criteria.

## VI. CONCLUSION

In this paper, we have presented a novel approach to curb detection. We have devised an unsupervised method that is applicable to various environment configuration and perspective views. We have demonstrated an application to a pedestrian robot and shown the robustness of our approach through a thorough experimental evaluation on real-world and synthetic data.

From a theoretical point of view, we have anchored our method to sound statistical models from the measurement process to the final inference tasks. This results into an elegant and efficient algorithm that is solely parameterized by sensor characteristics. Our approach reconstructs the environment as a mixture of plane segments and flags curbs at their

boundaries. For the parameter estimation, we have replaced the standard E-step of the Expectation-Maximization (EM) algorithm with loopy Belief Propagation (BP) and justified its utilization. Finally, we have tackled model selection issues with a graph-based segmentation heuristic.

As a future work, we envision to apply a fully Bayesian treatment to our method and investigate the use of Hierarchical Mixture of Experts (HME) [17], [18]. Additionally, we want to study the feasibility of a recursive estimation framework. Since BP is amenable to parallelization, it would also be beneficial to use a GPU implementation.

## ACKNOWLEDGMENT

This work has partly been supported by the EC under FP7-231888-EUROPA.

## REFERENCES

- [1] F. Oniga and S. Nedeveschi, "Polynomial curb detection based on dense stereovision for driving assistance," in *Proc. Int. IEEE Annual Conf. Intell. Transport. Syst. (ITSC)*, 2010.
- [2] J. Siegemund, D. Pfeiffer, U. Franke, and W. Forstner, "Curb reconstruction using conditional random fields," in *Proc. IEEE Intell. Veh. Sym. (IVS)*, 2010.
- [3] Y. Shin, C. Jung, and W. Chung, "Drivable road region detection using a single laser range finder for outdoor patrol robots," in *Proc. IEEE Intell. Veh. Sym. (IVS)*, 2010.
- [4] V. Pradeep, G. Medioni, and J. Weiland, "Piecewise planar modeling for step detection using stereo vision," in *Proc. Europ. Conf. Comput. Vis. (ECCV)*, 2008.
- [5] D. Yuan and R. Manduchi, "Dynamic environment exploration using a virtual white cane," in *Proc. IEEE Conf. Comput. Vis. Patt. Recog. (CVPR)*, 2005.
- [6] D. Angelov, B. Taskary, V. Chatalbashev, D. Koller, D. Gupta, G. Heitz, and A. Ng, "Discriminative learning of Markov random fields for segmentation of 3D scan data," in *Proc. IEEE Conf. Comput. Vis. Patt. Recog. (CVPR)*, 2005.
- [7] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis*. Chapman and Hall/CRC, 2003.
- [8] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.
- [9] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statistic. Soc., Series B*, vol. 39, no. 1, pp. 1–38, 1977.
- [10] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. Int. Conf. Machine Learning (ICML)*, 2001.
- [11] J. M. Mooij and H. J. Kappen, "Sufficient conditions for convergence of the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 53, no. 12, pp. 4422–4437, 2007.
- [12] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 498–518, February 2001.
- [13] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vis. (IJCV)*, vol. 59, no. 2, pp. 167–181, Sept. 2004.
- [14] J. M. Mooij, "libDAI: A free and open source C++ library for discrete approximate inference in graphical models," *J. Machine Learning Research*, vol. 11, pp. 2169–2173, August 2010.
- [15] S. Faria and G. Soromenho, "Fitting mixtures of linear regressions," *J. Statist. Comput. and Simul.*, vol. 80, no. 10, pp. 201–225, February 2010.
- [16] A. Rosenberg and J. Hirschberg, "V-Measure: A conditional entropy-based external cluster evaluation measure," in *Proc. Joint Conf. Empirical Methods in Nat. Language Processing and Comput. Nat. Language Learning (EMNLP-CoNLL)*, 2007.
- [17] M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the EM algorithm," *Neural Comput.*, vol. 6, no. 2, pp. 181–214, 1994.
- [18] C. M. B. M. Svendsen, "Bayesian hierarchical mixtures of experts," in *Proc. Conf. Uncertainty in Artificial Intell. (UAI)*, 2003.