

Received September 13, 2021, accepted September 23, 2021, date of publication September 24, 2021, date of current version October 4, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3115664

Ground Segmentation Algorithm for Sloped Terrain and Sparse LiDAR Point Cloud

VÍCTOR JIMÉNEZ¹, JORGE GODOY¹, ANTONIO ARTUÑEDO¹, AND JORGE VILLAGRA¹

Centre for Automation and Robotics (CSIC-UPM), 28500 Arganda del Rey, Madrid, Spain

Corresponding author: Víctor Jiménez (victor.jimenez@csic.es)

This work was supported in part by the Spanish Ministry of Science and Innovation through the National Project Programmable Systems for Intelligence in Automobiles (PRYSTINE) under Grant PCI2018-092928 and the National Project Product Security for Cross Domain Reliable Dependable Automated Systems (SECREDAS) under Grant PCI2018-093144, in part by the Community of Madrid through SEGVAUTO 4.0-CM Programme under Grant S2018-EMT-4362, and in part by the European Commission and Electronic Components and Systems for European Leadership (ECSEL) Joint Undertaking through the Project PRYSTINE under Grant 783190 and the Project SECREDAS under Grant 783119.

ABSTRACT Distinguishing obstacles from ground is an essential step for common perception tasks such as object detection-and-tracking or occupancy grid maps. Typical approaches rely on plane fitting or local geometric features, but their performance is reduced in situations with sloped terrain or sparse data. Some works address these issues using Markov Random Fields and Belief Propagation, but these rely on local geometric features uniquely. This article presents a strategy for ground segmentation in LiDAR point clouds composed by two main steps: (i) First, an initial classification is performed dividing the points in small groups and analyzing geometric features between them. (ii) Then, this initial classification is used to model the surrounding ground height as a Markov Random Field, which is solved using the Loopy Belief Propagation algorithm. Points are finally classified comparing their height with the estimated ground height map. On one hand, using an initial estimation to model the Markov Random Field provides a better description of the scene than local geometric features commonly used alone. On the other hand, using a graph-based approach with message passing achieves better results than simpler filtering or enhancement techniques, since data propagation compensates sparse distributions of LiDAR point clouds. Experiments are conducted with two different sources of information: nuScenes's public dataset and an autonomous vehicle prototype. The estimation results are analyzed with respect to other methods, showing a good performance in a variety of situations.

INDEX TERMS Belief propagation, channel-based, geometric features, LiDAR, Markov random field, obstacle-ground segmentation, sloped terrain, sparse point cloud.

I. INTRODUCTION

Autonomous driving is a challenging goal for which a reliable perception of the environment is essential. To that end, many automated vehicles are equipped with laser range scanners (LiDARs), as they provide a large amount of accurate 3D measurements. However, not all the information obtained is relevant; measurements relative to obstacles have to be distinguished from measurements relative to ground. This is crucial for typical perception tasks such as object detection-and-tracking or occupancy grid map generation. Simple height thresholding strategies have been used in some situations. Nevertheless, flat world assumption is rarely met, especially in driving environment where outdoor and uncontrolled

scenarios are commonly found. Indeed, slopes, road imperfections, vegetation and the complexity related with road users make this segmentation process much more complex than expected.

Several approaches for ground segmentation have been proposed in the literature. Dividing the point cloud into small groups (e.g. channels), and analyzing local geometric features inside them is a common strategy with proved good results. However, locally focused analysis is prone to noise in areas with few data. In contrast, different authors choose to model the ground height by analyzing the point cloud information as a whole and then perform the segmentation task comparing the points with the obtained estimation. The most frequent approximation is done by plane-fitting, but this approach has a strong dependency on a flat world assumption. Both sloped terrain and sparse data still motivate the research on

The associate editor coordinating the review of this manuscript and approving it for publication was Thomas Canhao Xu¹.

ground segmentation in point clouds. Less commonly seen approaches rely on Markov Random Fields (MRF) and Loopy Belief Propagation (LBP) for this task. These methods are less reliant of the plane ground condition, allowing local relieve changes, and have shown promising results in situations with sparse information [1], commonly found in LiDAR data. However, the approaches found are not strictly focused on ground segmentation, do not address sparse data or use simple geometric conditions to model the initial data cost.

The main contributions of this article are:

- The combination of a channel-based approach with a multi-label MRF approach, obtaining a novel strategy called Channel Based Markov Random Field (CB-MRF).
- The design of a channel-based ground segmentation algorithm, proposing additional rules and processes designed to overcome commonly found problems.
- The modeling of the data cost of the MRF in base of an initial estimation instead of local geometric features alone.
- The evaluation is carried out with real data from public datasets and from an automated vehicle prototype [2]. The proposed strategy is compared with similar approaches with qualitative and quantitative results.

The remainder of this paper is organized as follows. Section II provides an overview of related works and the different motivations that led to the chosen strategy. Section III presents the details of the strategy proposed, including the algorithm for initial classification and the calculation of the ground height map. Section IV gives some guidelines on how to parametrizate the proposed algorithm. Section V shows the experiments performed with two different sources of data. Finally, Section VI summarizes the presented work and proposes related future research areas.

II. RELATED WORK

The use of LiDAR for environment perception and situation understanding in Autonomous Vehicles (AVs) has significantly evolved in the last years (from [3] to [4]). Throughout this time, both feature-based ([5]) and grid-based ([6]) approaches for object detection have highlighted the need to differentiate ground impacts from object measurements. To that end, several approaches have been proposed for ground filtering in point clouds.

Plane-fitting and line-fitting has been widely tested. Authors in [7], [8] use a RANSAC algorithm to fit points into a single plane. However, ground is rarely flat, thus the assumption of a single ground plane leads to false-classifications. To overcome this issue, similar approaches divide the scene in different sectors for which local planes or lines are estimated (e.g. [9]–[14]). The flexibility of plane fitting is increased with these techniques, but on the one hand, wide sectors still struggle fitting all points inside a single local plane and, on the other hand, small sectors may suffer from a local lack of points. Some additional techniques can be used to improve the inliers classification, such as the point-wise

normal [15] or the tangent to the curves produced by the sweep of the laser beams [16].

A different strategy consists on analyzing local geometric features from neighbouring points, such as height differences, gradients or normal vectors. The elevation maps commonly used in DARPA Urban Challenge [17] divide the point cloud into cells and compute the maximum height difference within them. If this difference exceeds a threshold value, all points within the cell are considered as obstacles. This simple approach is able to handle irregular terrains, but its simple feature criteria may produce wrong classifications in cells with few points or in cells which contain overhanging structures. Reference [18] computes the point's normal vector and a quality value using the 8-neighbouring points so that the point is a ground candidate if the obtained normal vector is similar to the ego-vehicle's upward edge. Afterwards, candidates are grouped using a distance based clustering algorithm, and large clusters are classified as ground. This method uses a more complex feature analysis, but may suffer in scenarios with heavy slopes or regions low density of points. Reference [19] takes advantage of the multi-layer LiDAR sensors common operation pattern and analyses the relationship between points from different layers, but with the same azimuth angle (in the vertical direction); this type of approaches are commonly known as channel or scan-line based algorithms. Assuming that these points may behave similarly unless obstacles appear, each point inherits the classification of its previous point. Certain geometric conditions of gradient, distance and height trigger the change from ground to obstacle or vice versa. This strategy adapts correctly to ground height changes, thus is able to correctly classify scenarios with sloppy terrain and vegetation, but ignores the relationship between angular adjacent points and is strongly dependant of parameters configuration. An enhanced version of this strategy was proposed in [20], where vertical and horizontal directions are analyzed an combined together. Nevertheless, it is still highly dependant on geometric parameters and does not allow the joint analysis of multiple LiDAR sensors. Similar approaches that divide the point cloud in smaller groups and classify points following their relationships can be found, [21], [22].

Other authors use these local features too, but perform additional processes in order to take into account the point's context information. Reference [23] applies a channel-based algorithm for initial classification and then improves this estimation by taking into account the inter-relationship between channels. Initial ground points are projected into a grid map where each cell stores the mean height. Then, a median filter is applied smoothing the height in the grid map and filling some of the empty cells. Finally, points are re-classified in according to their height over the cells' ground height. Reference [24] also use the same strategy to correct the estimation achieved with [10]. This grid-based representation and smoothing process corrects occasional false-classifications produced during the initial classification and relaxes its parameter dependency. However, the points density highly

TABLE 1. Comparison of ground extraction approaches.

	Plane/line fitting	Local features	Enhanced channel-based	Higher order inference	Deep learning
Performance in general scenarios	++	+	++	+++	+++
Performance in sloped terrain	0	++	++	+++	+++
Performance with sparse data	+	0	+	+++	++
Simplicity	++	+++	++	0	0
Parameter independency	++	0	+	+	+
Dataset independency	+++	+++	+++	+++	0

affects this enhancement step, leading to false-classifications in farther distances, where point cloud data is sparse. Higher order inference algorithms have been proposed in order to handle sparse point cloud situations. References [25], [26] use MRF with Belief Propagation algorithms (BP) in order to classify navigable regions, demonstrating that the data propagation can successfully handle low point density areas. Inspired by them, [1] proposed a ground segmentation method based on multi-label MRF and LBP, where the data cost is fed by height difference and potential occlusions. This strategy obtains a robust segmentation in rough terrain and compensates the sparse point cloud distribution. However, its outcome is dependant of the single height condition which, as stated previously with elevation maps, struggles with overhanging structures and lack of points. Other representative techniques with higher order inference algorithms can be found, for example, in [27] which models the point cloud as a MRF, in [28], which uses Gaussian Process regression (GP), or in [29], which uses Conditional Random Fields (CRF) and extends the spatial dependencies to spatio-temporal dependencies.

Approaches based on deep learning [30], [31] use Convolutional Neural Networks (CNN) to identify obstacles in the point cloud data. These techniques can take into account features that can be hardly modeled by traditional methods. Nevertheless, large amounts of data have to be used for training and good results may be limited to scenes with similar conditions than the training set.

Finally, it is worth considering some airborne LiDAR based works. Despite the different field of application, some authors have used airborne LiDAR approaches in autonomous vehicles applications. For example [29], mentioned above, was inspired by [32], which proposed in turn a new hybrid CRF model to extract Digital Terrain Model (DTM) and [33] applied the Cloth Simulation Filter (CSF) [34] to detect rocks on the road.

Table 1 shows a high-level summary of the different families of techniques with regards to some key features for ground extraction.

Motivated by the analysis of previous methods, this article proposes a strategy composed of two steps:

- 1) Initial point cloud segmentation: Inspired by [19], [20], [23], points are clustered by azimuth angle and analyzed in an ordered way from close to far, the classification of each points depends on its previous points and different geometric features. Two of these mentioned works already introduced the necessity of a posterior

filtering step to correct noise. However, the proposed methods are not adequate for regions with sparse information.

- 2) Ground height modeling with final point cloud segmentation: Following the strategy proposed by [1], [25], [26], the surrounding ground height is estimated using a multi-label MRF and LBP. As a result, data is propagated along nodes to obtain a good ground estimation even in areas with reduced information. The final point cloud classification is obtained with respect to this ground height map.

III. GROUND SEGMENTATION STRATEGY

The CB-MRF method is divided in two main steps: (i) an initial channel-based classification and (ii) the final classification based on MRF and LBP. Fig. 1 displays a general view of the algorithm, whose main stages are described below.

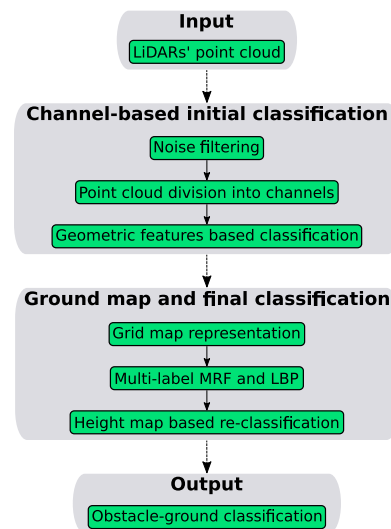


FIGURE 1. Ground segmentation algorithm scheme.

A. CHANNEL-BASED INITIAL POINT CLOUD CLASSIFICATION

LiDAR sensors used in autonomous driving tasks usually have multiple layers. These layers commonly share the same horizontal Field of View (FOV), but have different pitch angle. Laser beams from different layers can thus be directly clustered by azimuth angle or capture time, obtaining a sorted sequence of points, each one from each layer, with increasing distance over the plane proportionally to the layer's

pitch angle, i.e. typical concentric point rings around the ego-vehicle. Henceforth, these clusters will be referred as channels.

Under the assumption that the ground height may not heavily vary between subsequent layers' scans, ground and obstacle points are identified. Thus, points inside a channel are analyzed in an ordered way, from the closest to the farthest. An additional virtual point is located under the sensor, at the theoretical ground height and it is always classified as ground. During the analysis, points can get three possible labels: *obstacle*, *ground* or *doubt*, but, at the end, all the points of the channel have to be classified either as *obstacle* or *ground*.

As shown in Fig. 1, a noise filtering task is performed previously. During this first step, points considered as outliers or without valuable information receive an additional label *noise* and are ignored during posterior steps. The process for filtering the noise is explained in Section III-C1.

In the following, the algorithm to classify points inside a channel and the heuristic rules applied are explained.

1) ALGORITHM DESCRIPTION

Algorithm 1 shows the pseudo-code for classifying points inside a channel. Note that p refers to a point inside the channel \mathcal{C} and l to the point's classification label. Subscripts i refer to the current analyzed point, g to the last point classified as ground and 0 to the virtual point. As shown, each point is classified based on its previous points:

- If the previous point is *ground*, the algorithm seeks for obstacle evidences (*CheckObstacle*). If obstacle conditions are met, the point is either classified as *obstacle* or *doubt*. Otherwise, it inherits the *ground* classification.
- If the previous point is *obstacle*, the algorithm seeks for ground evidences (*CheckGround*). If ground conditions are met, the point is classified as *ground*. If not it inherits the *obstacle* classification.
- If the previous point is *doubt*, the algorithm seeks for evidences of both, obstacle and ground (*CheckBoth*). If obstacle conditions are met without doubt, the current point and all previous doubt points are classified as *obstacle* (*CorrectDoubtPoints*). An analog reasoning applies for the case that ground conditions are met. If none of these two situations happen, the current point inherits the *doubt* classification.
- If the current point is considered as noise, the algorithm ignores it.

2) HEURISTIC RULES

As just explained, each point classification relies on its previous points. The relationships between these points are analyzed in order to determine whether a new classification is assigned or the previous one is inherited. This analysis is empirically based on different geometric conditions that can be inferred from common point cloud distributions, the LiDAR functioning and the considered application. Therefore, different heuristic rules are defined, that are

Algorithm 1 Channel Ground Segmentation

Input: Channel's points

Output: Obstacle-ground labels

```

 $l_0 \leftarrow \text{Ground}$ 
for all  $p \in \mathcal{C}$  do
  if CheckNoise( $p$ ) then
    continue
  end if
  switch  $l_{(i-1)}$  do
    case Ground
       $l_i \leftarrow \text{CheckObstacle}(p_i, p_{(i-1)}, p_g)$ 
    case Obstacle
       $l_i \leftarrow \text{CheckGround}(p_i, p_{(i-1)}, p_g)$ 
    case Doubt
       $l_i \leftarrow \text{CheckBoth}(p_i, p_{(i-1)}, p_g)$ 
      if  $l_i \neq \text{Doubt}$  then
        CorrectDoubtPoints( $l_i$ )
      end if
    end switch
  end for

```

described below and showcased with representative examples in Fig. 2.

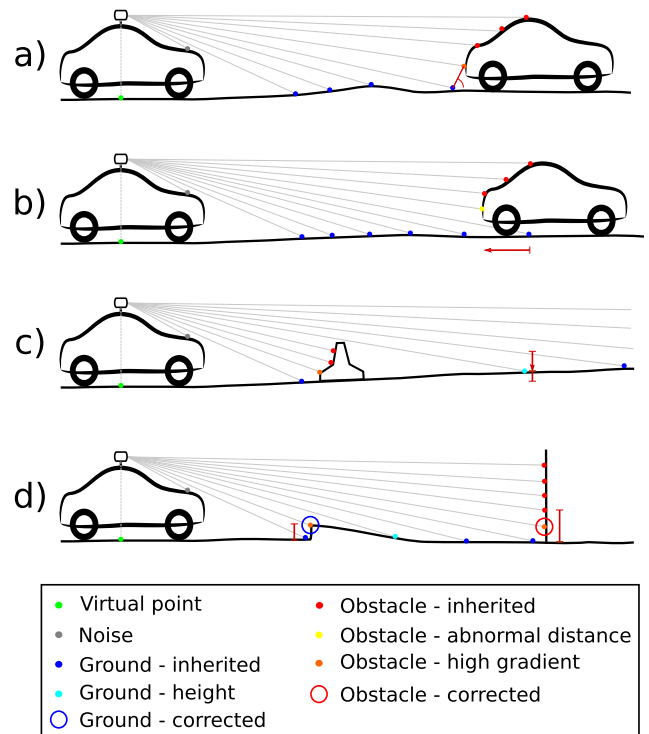


FIGURE 2. Illustrative examples for obstacle-ground classification.

Two heuristic rules are defined to find obstacle evidences. If anyone of them is met, the current point can be classified as *obstacle*:

- Maximum slope allowed: if the gradient between the current point $\mathbf{p}_i = (x_i, y_i, z_i)$ and the previous one

$\mathbf{p}_{(i-1)} = (x_{(i-1)}, y_{(i-1)}, z_{(i-1)})$ exceeds a threshold α_θ , the current point is considered to belong to an obstacle:

$$\arctan \left(\frac{z_i - z_{(i-1)}}{\sqrt{(x_i - x_{(i-1)})^2 + (y_i - y_{(i-1)})^2}} \right) > \alpha_\theta \quad (1)$$

The use of the gradient is selected because the ground may have small slopes. Moreover, since point cloud data becomes sparser as distance to the sensor increases, it is common to detect obstacles as heavy slope changes and not as vertical structures. An example is shown in Fig. 2 a).

- **Abnormal distance:** if the current point $\mathbf{p}_i = (x_i, y_i, z_i)$ is closer to the sensor than the previous one $\mathbf{p}_{(i-1)} = (x_{(i-1)}, y_{(i-1)}, z_{(i-1)})$, in terms of radial distance, the current point is considered to belong to an obstacle:

$$\sqrt{x_i^2 + y_i^2} < \sqrt{x_{(i-1)}^2 + y_{(i-1)}^2} \quad (2)$$

As explained before, if all laser beams hit on the ground, points are sorted by distance, and a disturbance in this order may mean the presence of an obstacle. An example of this situation is shown in Fig. 2 b).

These two obstacle rules can be triggered by ground imperfections or small objects which do not provide valuable obstacle information, such as bumps, grass or curbs. In order to avoid this false-classification, obstacles have to be confirmed:

- **Relevant height difference:** if the height between a tentative obstacle point $\mathbf{p}_i = (x_i, y_i, z_i)$ and the last ground point $\mathbf{p}_g = (x_g, y_g, z_g)$ exceed a certain threshold α_h , the point is considered as *obstacle*, otherwise it is classified as *doubt*:

$$\begin{cases} \text{obstacle,} & z_i - z_g \geq \alpha_h \\ \text{doubt,} & \text{otherwise} \end{cases} \quad (3)$$

Fig. 2 d) shows an example of a point corrected to ground and a point validated as obstacle.

Likewise, three heuristic rules are defined to find ground evidences. In contrast to the obstacle rules, the three of them have to be simultaneously met in order to consider a point as *ground*.

- **Expected distance:** the current point has to be farther than the last ground point, satisfying the second obstacle rule.
- **Height reduction:** the height of the current point $\mathbf{p}_i = (x_i, y_i, z_i)$ has to be lower than the height of the previous point $\mathbf{p}_{(i-1)} = (x_{(i-1)}, y_{(i-1)}, z_{(i-1)})$:

$$z_i < z_{(i-1)} \quad (4)$$

This may mean that the obstacle is over and the beam is hitting again on the ground.

- **Similar height to last ground point:** the height of the current point $\mathbf{p}_i = (x_i, y_i, z_i)$ has to be similar or lower than the height of the last ground point $\mathbf{p}_g = (x_g, y_g, z_g)$:

$$z_i - z_g < \alpha_h \quad (5)$$

The last ground point is the strongest reference of the ground available. In this case, the height relationship is chosen instead of the gradient because long distances may easily appear between the last obstacle point and the new ground point, thus gradient thresholding would allow high height differences. Fig. 2 c) shows an example.

Finally, in order to control wrong *obstacle* confirmation by long slopes or the ending of the channel, if *doubt* points are not solved before a certain distance or the end of the channel, the obstacle evidences are assumed not to be strong enough and, therefore all *doubt* points are corrected to *ground*.

B. GROUND HEIGHT MAP ESTIMATION AND POINT CLOUD FINAL CLASSIFICATION

The initial obstacle-ground segmentation relies on clustering the points into small groups. This division provides different advantages, but it also constrains the information used, ignoring valuable dependencies between channels and making the estimation susceptible to local errors.

Dependencies between adjacent channels can be easily addressed with different methods, but obtaining a good estimation in areas with few point cloud data is a harder task. To meet this objective, the elevation of the ground surrounding the ego-vehicle is represented as a ground height map. This map is modeled in terms of a MRF and solved using LBP algorithm [35]. Thereafter, every point receives a final obstacle-ground classification by comparing its height with an estimated ground map.

1) GRID REPRESENTATION

During this step, data is represented as a polar grid map. Each cell defines a portion of the surrounding area in terms of angle and radial distance ($\Delta\theta$, Δr). The data relative to the points' height and initial classification is gathered inside these cells. Points classified as noise are not included.

2) MRF-BASED MULTIPLE LABELING DESCRIPTION

Let $G = (V, E)$ be an undirected graph with nodes $v_i \in V$ and edges $(v_i, v_j) \in E$ which describe the dependency between adjacent nodes v_i and v_j . A finite set of labels L defines the possible height values that nodes can take and a labeling function f assigns a label $f(v_i) \in L$ to each node $v_i \in V$. This graph is represented by the polar grid previously introduced. Each node of the graph corresponds to a certain cell of the grid and the dependencies between nodes are set to the four-connected neighbouring cells: forward, backward, clockwise and counterclockwise. Similarly, labels refer to height intervals (Δh) within which the ground height may be found.

The quality of a label is calculated by the energy function:

$$\mathcal{E}(f) = \sum_{v_i \in V} D(f(v_i)) + \sum_{(v_i, v_j) \in E} W(f(v_i), f(v_j)) \quad (6)$$

where $D(f(v_i))$, normally referred as data cost, is the cost of assigning the label $f(v_i)$ to the node v_i and $W(f(v_i), f(v_j))$,

normally referred as smoothness cost, is the cost of assigning the labels $f(v_i)$ and $f(v_j)$ to the neighboring nodes v_i and v_j . The details of these 2 costs are given in the following 2 sub-sections.

The computation of the optimal labeling that minimizes the energy in (6) is performed using an implementation of the LBP algorithm (c.f. [35], [36]). Briefly explained, the LBP algorithm passes messages along nodes in the four possible directions. Each message consists of a vector with size equal to the number of possible labels. After a certain number of iterations, the belief vector for each node is computed and the label with maximum belief (minimum cost) is selected as optimal label. Readers may refer to [1], [26] for LBP implementations in similar applications.

Finally, all points are classified comparing their height with the ground height estimated for their node:

$$l_i = \begin{cases} \text{ground}, & z_i^c - z_m^c < h_g \\ \text{obstacle}, & \text{otherwise} \end{cases} \quad (7)$$

where, l_i and z_i are the obstacle-ground label and the height of i th-point in the cell c , z_m^c is the height relative to the cell's optimal label and h_g is the height difference parameter.

3) DATA COST MODELING

The data cost defines the cost for the node v_i to take the label $f(v_i)$. Its value is defined depending on the points within the cell (node). Recall that labels refer to possible ground height values and the most probable label has the lowest cost. Similarly to [1], three different data cost functions are defined depending on the points distribution and their initial classification:

- Category 1 is defined for cells without points inside it. In this situation the point cloud does not provide any information for the current node. Thus, ground can be found at any label, so every label is equally probable:

$$D(f(v_i)) = 0 \quad (8)$$

- Category 2 is defined for cells with one or more points inside it and at least one of them has been initially classified as ground. Under this circumstance, the ground height is probably located at the label with the highest number of points initially classified as ground. In case there is a tie, the label with the lowest height is selected. The cost of the other labels is defined with respect to their proximity to this label. Therefore, the data cost can be defined as a truncated linear function such as:

$$D(f(v_i)) = \min(|f(v_i) - g(v_i)|, \tau) \quad (9)$$

where $g(v_i)$ is the reference label selected as the most probable label (highest number of ground points) and τ is a truncation value. The truncation value is a common tool introduced to make the cost function more robust, e.g. against high differences between adjacent nodes or $g(v_i)$ wrongly selected due to false-classifications.

- Category 3 is defined for cells with one or more points, but none of them has been initially classified as ground. This situation may happen for different reasons, such as non-vertical obstacles (Fig. 2), occlusions or false-classifications. Therefore, the ground height is probably located at or under the point with the lowest height inside current cell.

$$D(f(v_i)) = \begin{cases} \min(f(v_i) - g(v_i), \tau) & f(v_i) > g(v_i) \\ 0 & f(v_i) \leq g(v_i) \end{cases} \quad (10)$$

where $g(v_i)$ is the reference label selected as the label with the point with the lowest height and τ is the truncation value.

4) SMOOTHNESS COST MODELING

The smoothness cost defines the cost of assigning the labels $f(v_i)$ and $f(v_j)$ to the neighboring nodes v_i and v_j . As already stated on Section III-A, the ground height may not vary abruptly, i.e. labels should vary slowly too. This is an assumption commonly made in similar problems [36] and is typically modeled by the truncated linear function:

$$W(f(v_i), f(v_j)) = \min(s|f(v_i) - f(v_j)|, \rho) \quad (11)$$

where s is a rate to weight the cost and ρ is a truncation value in order to allow discontinuities.

C. NOISE FILTERING AND IMPROVING STEPS

This section explains three refinement steps which are included in different sections of the CB-MRF strategy to improve the estimation obtained with real-data but are not considered as part of the main algorithm.

1) NOISE FILTERING

As shown in Fig. 1, the first step consist on filtering noise points. In order to identify these points two methods are applied:

- Safe distance thresholds: Points with irregular heights - for example more than 5 meters under the ego-vehicle's height, are considered as noise. Similarly, points hitting on the ego-vehicle do not provide valuable information, thus points within the area covered by the ego-vehicle are also considered noise
- Simple plane-fitting: During practical experimentation, some noise points under the ground plane have been found, being the most harmful those close to the ego-vehicle. In order to filter these points, it is assumed that the patch of ground over which the ego-vehicle is located is relatively flat, thus close ground points may approximately fit on a plane at the theoretical ground height. Any plane fitting method can be used for this purpose; for example computing the normal vector using eigenvectors. Therefore, close noise points are identified by:

$$(\mathbf{n}^{cl})^T \mathbf{p}_i + d^{cl} < h_n \quad p_i \in P^{cl} \quad (12)$$

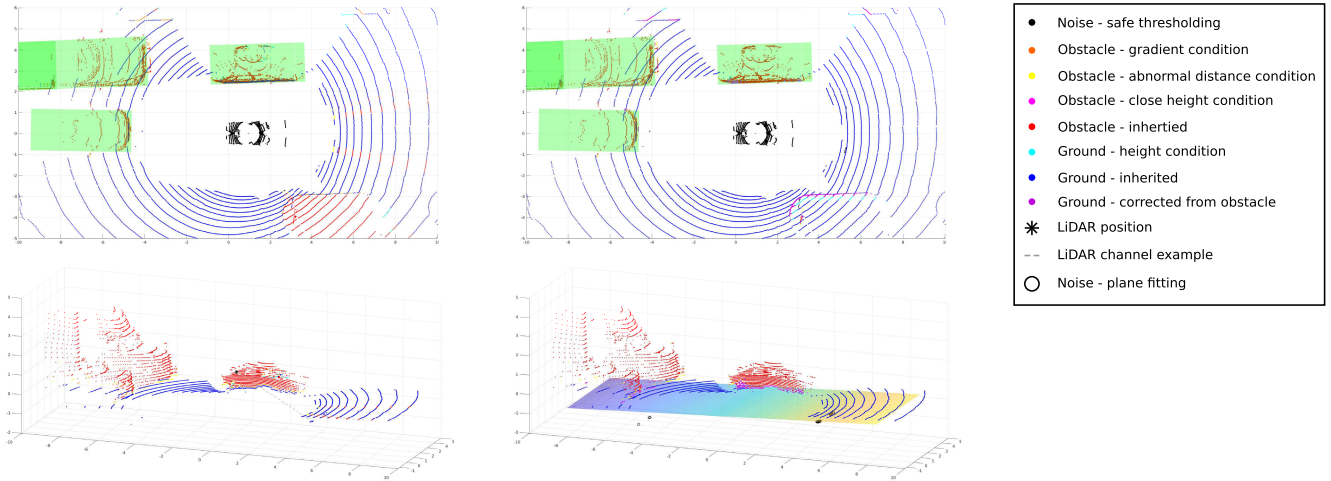


FIGURE 3. Left: Result of applying a ground segmentation algorithm based on [19]. Right: Result of applying the proposed channel-based algorithm.

where P^{cl} is the set of points close to the ego-vehicle and the ground theoretical height, n_c and d_c are the plane coefficients and h_n is a height safety threshold. Yet, assuming that the ego vehicle is located over a flat patch is a strong assumption, thus it is only applied if the number of points classified as noise does not exceed a threshold n_n .

2) ADDITIONAL HEURISTIC RULE

In section III-A2, two rules to identify obstacles inside a channel were stated. However, when using certain sensor configurations, such as the LiDAR sensor mounted on top of the vehicle, the distance between the sensor and first ground points (inner ring) takes values from two to five meters approximately, see Fig. 3. This may produce false-classifications when not sharped obstacles enter inside the inner ring, since the gradient condition allows a height difference proportional to the distance. Using the same assumption made in Section III-C1, the ground inside the inner ring should not vary abruptly. This statement introduces a new special heuristic rule by which points found inside the area defined by the inner ring (P^r) are classified as obstacles if they exceed a certain height threshold α_r :

$$z_i - z_0 > \alpha_r \quad z_i \in P^r \tag{13}$$

where z_0 is the theoretical ground height under the sensor.

3) ENHANCEMENT OF VERTICAL STRUCTURES

In the last step, named as “correct classification” in Fig. 1, the initial classification is corrected with respect to the estimated ground height map. As explained in [23], applying a height threshold over the estimated plane as proposed in (7) may lead to some false-classifications in the lower parts of vertical structures. To overcome this issue [23] compares the number of ground points and non-ground points inside each cell, considering a vertical structure inside the cell is assumed

if the number of non-ground points is larger. This simple approach, while useful in some situations may fail in others, as in overhanging structures. Instead, it is proposed to identify cells with potential vertical structures as those with at least n_v contiguous labels with points inside them (similarly to the concept applied on [1] to remove overhanging points). Additionally, since the initial classification properly performs in this task, only the label from ground to obstacle of those points that were initially classified as obstacle is changed.

IV. PARAMETRIZATION GUIDELINES

This section is intended to provide a parametrization guidance for potential users of the CB-MRF algorithm.

The first step, the channel-based algorithm, is based on heuristic rules and computes an initial classification, thus their values can be loosely defined using pragmatic knowledge. It depends on two main parameters and models both model the surface inclination and smoothness:

- Gradient threshold (α_θ): it models the possible surface slope. Flat scenarios can be handled with low values, for example 15 degrees. On the contrary, sloped scenarios require higher values, e.g. [20] used 30 degrees for urban scenarios and [19] used 45 degrees for mountain roads.
- Ground height threshold (α_h): it models a significant height difference with respect to the ground, i.e. the height of the smallest relevant obstacles. In urban scenarios this value may be higher than curbs or bumps, around 20 – 30 centimeters.

The second step, ground height map estimation, depends on different parameters; some of them can be deterministically set, while others may be more empirical:

- The grid size is defined to cover sensors FOV and the possible ground height variation. Cell size is related to the ground smoothness - the bigger the cell, the lower height variation is expected. For example, for urban scenarios, values around 20cm and 2 deg have shown good results.

- Smoothness cost rate (s) and truncation values (τ , ρ): are related to the cost of assigning a specific label and should be experimentally adjusted, some references can be found in [1], [26].
- For the message passing, the scheme forward - clockwise - backward - counterclockwise has shown the best results. Additionally, since the data cost of the MRF is already based on an initial classification, convergence is fastly achieved, thus few iterations are needed, e.g. 5 or less.
- Height map's ground threshold (h_g): is a common security threshold that models the possible error of the estimated ground height map. For urban scenarios, [16] used a height value of 20cm and [23] used a double security threshold: 10cm for the obstacle-ground classification and 25cm for an additional curb label.

From all these parameters, the most influential ones are: h_g that is responsible for over/under-segmentation and α_h that is the main tool for obstacle detection.

With respect to the noise filtering techniques and refinements steps suggested on Section III-C, the parametrization highly depends on the sensor's performance and location. Therefore, each user should evaluate its own necessities.

V. EXPERIMENTAL RESULTS

This section describes a set of experiments conducted to validate the feasibility of the proposed strategy.

A. EXPERIMENTAL DATASETS

The work presented on this paper has been tested with two datasets: the one obtained using an automated vehicle prototype (AUTOPIA) and the open-access and well-known repository nuScenes [37].

- AUTOPIA: the prototype counts with a set of different proprioceptive and exteroceptive sensors. In this work, the two VLP16 mounted on its roof are used. These sensors cover 360deg around the vehicle, but they are tilted in order to capture more information in the front and sides of the vehicle.
- nuScenes: public dataset which provides exhaustive data from typical autonomous driving scenarios. Data is collected in Boston and Singapore, two cities with dense traffic and challenging driving situations. In this work the information of the single LiDAR of 32 layers is used. This sensor covers 360deg and is mounted horizontally on top of the vehicle. In addition to the LiDAR raw data, every point has an associated semantic label, e.g. vehicle, flat, etc. which is used as ground truth.

The validation with two different types of datasets allow us to show the results obtained for different sensors models and different location configurations. Moreover, AUTOPIA's prototype permit to demonstrate the capability of fusing two sensors. In addition to that, the ground truth provided by nuScenes dataset allows qualitative validation.

B. IMPLEMENTATION DETAILS

Four different segmentation approaches have been implemented, tested and evaluated:

- The initial classification explained in Section III-A alone, hereunder named as channel-based algorithm.
- The channel-based algorithm proposed with a median filter as posterior filtering step to improve the estimation. Initial ground points are translated into a polar grid map as the mean height value for each cell, a median filter is then applied over this grid and all points are finally reclassified with respect to their height over the obtained height grid map.
- The algorithm of [1], based on height differences, using MRF and LBP.
- The full strategy proposed in this article: channel-based algorithm with the MRF-based to improve the estimation.

The parametrization of the different steps of the CB-MRF is as follows:

- Noise filtering: $h_n = 50cm$, $n_n = 1\%$. The rectangular path is defined slightly bigger than the inner ring: 16×10 meters.
- Channel-based algorithm: $\alpha_\theta = 20$ degrees, $\alpha_h = 20cm$, $\alpha_r = 50cm$.
- MRF: The grid size is defined to cover 360 deg and a distance up to 60 meters, LiDAR sensors used in this work barely capture significant information at farther distances. Possible height labels have a range from -2.5 to 4.5 meters. Cell size is experimentally set to 20 cm and 2 deg. $\tau = 5$, $s = 0.5$ and $\rho = 3$.
- LBP: Scheme forward - clockwise - backward - counterclockwise and 5.
- Final classification: $h_g = 10cm$

The parameters for the median filter (MF) are empirically set, obtaining good results for the same grid configuration as the one used for the CB-MRF strategy: [20cm, 2 deg] and $h_g^{MF} = 10$ cm.

The algorithm explained in [1] is tested with the configuration specified in that article.

C. QUALITATIVE RESULTS

A qualitative evaluation is performed in order to explain the advantages and motivations of the CB-MRF algorithm. For this task, both datasets are used, the one obtained from AUTOPIA's autonomous vehicle and nuScenes datasets. First, the channel-based algorithm is individually analyzed. Then, the full proposed strategy (CB-MRF) is analyzed and compared with other similar approaches.

1) CHANNEL-BASED GROUND SEGMENTATION

Despite its simplicity, the channel-based ground segmentation technique is able to achieve good results. Nevertheless, there are some situations in which wrong classifications can appear. Fig. 3 shows an example from the nuScenes dataset, where a simple channel-based algorithm

(left column) achieves a good performance overall, but it also fails in three challenging situations: (i) high curbs; (ii) noise under the ground plane; and (iii) a close vehicle that crosses the first ground scan.

The left picture shows the result of applying the algorithm presented in [19], while the right plot shows the initial classification obtained when applying the proposed channel-based algorithm. Vehicles are represented by their ground truth boxes in green. The estimation obtained from the point cloud is represented in different colours: reddish points represent obstacles, bluish points represent ground and black points represent noise. In the left picture, it can be noticed how the vehicles behind the ego-vehicle are perfectly segmented, even ground points under them. On the contrary, a big ground area on the front-right side of the ego-vehicle has been wrongly classified as obstacle. This area corresponds to a sidewalk with a high curb, where two layers hit, leading to 90 deg gradients estimation. Recall measuring high gradients is one of the conditions to classify points as obstacle. Moreover, since the height of farther points within the same channels do not decrease, all of them inherit the obstacle label. Following the same reasoning, the noise found in front of the ego-vehicle and under the real ground plane, produces a wrong classification that is inherited by all posterior points. Finally, the vehicle inside the inner ring has some points wrongly classified as ground. This wrong classification is consequence of the LiDAR location, as explained in Section III-C2. In this case, the distance between the LiDAR and the vehicle is higher than 2 meters, thus the gradient analysis allows high height differences too. This may be a negligible error for obstacle detection since the footprint of vehicle is detected almost perfectly, but for the posterior ground height estimation step, it provides a wrong reference that can harm the result.

In order to deal with these three situations, three modifications or additional steps have been proposed for the channel-based algorithm:

- To avoid wrong classifications due to abrupt ground changes, the height difference between consecutive points is compared with a certain height threshold. If it does not exceed this threshold, the point is firstly classified as *doubt* and its final classification is defined considering the following points.
- In order to filter noise points close to the ego-vehicle and under the ground plane, the ground near the ego-vehicle is approximated as a local ground plane leading to classify points under this local plane as noise.
- To minimize the wrong classification of obstacles inside the inner ring, the proposed version does not only analyse the gradient but also constrain a maximum height change with respect to the theoretical ground height under the ego-vehicle.

Fig. 3 right column displays the results obtained when applying these changes. It can be seen how the sidewalk is now correctly classified, the wrong classification produced by the noise under the ground has being avoided, and the

classification of the lower points of the vehicle inside the inner ring has been improved.

2) COMPLETE GROUND SEGMENTATION STRATEGY (CB-MRF)

This section presents the results obtained with the complete strategy. First, the improvements achieved with respect to the channel-based algorithm alone (Section III-A) are shown. Then, a comparison with a basic plane fitting algorithm is performed in order to expose the ability of the CB-MRF strategy to handle sloped terrains. Finally, two urban scenes are use to highlight the advantages obtained with respect to the previously mentioned similar strategies.

Fig. 4 shows the results of two scenes collected with AUTOPIA's automated vehicle prototype. The point cloud is coloured based on the estimation output, being red for obstacles and black for ground. Vehicles' position and shape, drawn in green, have been manually labeled. The sensors setup of this prototype is useful to demonstrate that the ground height map modeling is able to fuse the information from various LiDARs, an issue that the channel-based algorithm is unable to handle by design. The scene shown in the first column illustrates how high grass areas cause a complicated situation for the channel-based algorithm. Since they are highly irregular but low dense, some grass points are classified as obstacles and areas beneath inherit this obstacle condition. When applying the MRF and the LBP, the information is analyzed as a whole and the surrounding ground height is approximately estimated, thus most of the initially misclassified points are corrected. The second column shows a complex situation where the proposed channel-based algorithm is able to handle the occlusion on the right side but not on the left. Additionally, the vehicle in front of the ego-vehicle is not detected. This misbehaviour is due to the fact that the channel-based algorithm estimates ground for each point cloud separately. This effect, together with the sparse information results in a total miss-detection of the vehicle. It can be seen that when the ground height is modeled, the estimation highly improves. On the left side, the vehicle is clearly distinguished from the ground and the detection of the vehicle in front is enhanced, as more than half of the points are correctly classified.

The proposed approach is based on the analysis of different strategies, but especially on two algorithms specialized in handling sloped terrains. Fig. 5a shows the result of a scene with a high slope variation. Fig. 5b shows the result of a simple plane fitting approach, proving that a single plane cannot properly solve these situations. On the contrary, Fig. 5c shows that the proposed approach is able to handle heavy slopes.

The following scene, represented in Fig. 6, is especially selected to easily display the advantages of: (i) using MRF and LBP rather than simpler local filters which cannot propagate information; and (ii) using the estimation of an initial ground segmentation algorithm to model the data cost of the MRF instead of just using height differences between points of the same cell which cannot correctly model obstacles with

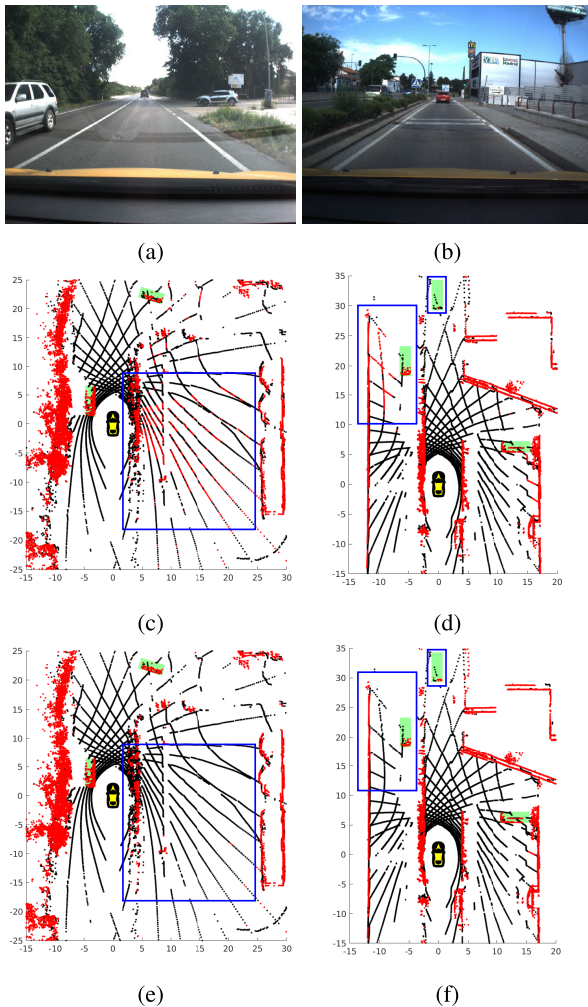


FIGURE 4. Two scenes with common issues found in channel-based approaches. Left column: low dense vegetation, right column: occlusions. Comparison between the initial classification and the final classification. a), b) Picture of the scene in front the ego-vehicle. c), d) Initial classification. e), f) Final classification.

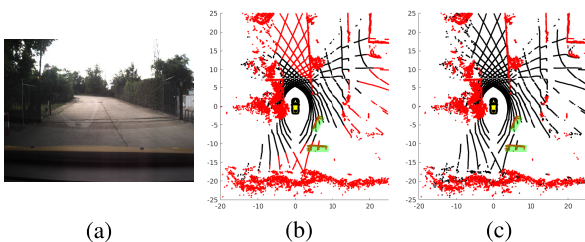


FIGURE 5. High sloped terrain. Comparison between a plane-fitting method and the CB-MRF algorithm. a) Picture of the scene in front the ego-vehicle. b) Plane-fitting method. c) CB-MRF algorithm.

few points. The scene is a common urban scene, extracted from nuScenes dataset, where the vehicle highlighted with a blue rectangle is only detected by a single layer of the LiDAR sensor.

The result of the four approaches is displayed in the different pictures of the figure: a) our proposed channel-based

algorithm, b) our proposed channel-based algorithm with median filter, c) the approach of [1], based on MRF and LBP, and d) the completely proposed strategy, the CB-MRF algorithm. It can be noticed how the four versions produce a good classification in general terms, but the detection of the highlighted vehicle varies. It is drawn in green if it is counted as detected and in red if not. The channel-based segmentation algorithm by itself is able to identify some of the points hitting on the vehicle as non-ground due to the gradient condition. From this estimation, if the median filter is applied, the vehicle detection is lost. This happens because the vehicle is not close enough to truly ground points, thus the ground height of this area is estimated just considering the false-classifications made by the initial estimation. This leads to a worse correction, since ground is wrongly estimated at the height of the points belonging to the vehicle. In the case of the algorithm proposed in [1], the vehicle is neither detected. Since only one layer hits on the vehicle the height difference condition to detect obstacles is not triggered. Additionally, there are no occlusions between this vehicle and the ego-vehicle. Therefore, the cells that gather vehicle’s points are modeled as potential ground for the data cost. From this wrong categorization, the LBP is not able to converge over the correct ground labels. On the contrary, when applying the strategy proposed by this work, the data cost receives a better description of the vehicle, some points are wrongly classified as ground but others not. As a result, the LBP is able to better estimate the ground height and most of the points belonging to the vehicle are correctly classified.

Fig. 7 shows a more complex scene from the dataset of [37]: an urban parking with several vehicles. Different obstacles can be found, and among them there are vehicles with low height and, in some cases, few points. Detected vehicles are coloured in green, undetected in red and unperceived by the LiDAR (too few laser hits) in grey. Again, the four approaches are tested and, in general, the performance is good for all of them. Obstacles are correctly segmented from ground points, even distinguishing ground under vehicles. However, depending on the approach some of the vehicles are confused with ground reliefs. These vehicles are mainly found in regions where point cloud data is sparse. The reasoning applied in the previous example applies to this case too, as the combination of the channel-based algorithm with the multi-label MRF performs better than the other strategies tested.

The vehicle inside the small blue rectangle is not detected by the initial channel-based segmentation algorithm. This is because the distance to the previous layer hitting the ground is long, allowing high height differences by the gradient condition. The other three approaches, on the contrary, are able to correctly identify this vehicle, since few points of a second layer hit the vehicle at the bottom. These lower points guide the height grid map estimations and, hence, vehicle’s upper points are correctly corrected to obstacle points. The big blue rectangle highlights an area with four vehicles and sparse information. Most of the points hit on the

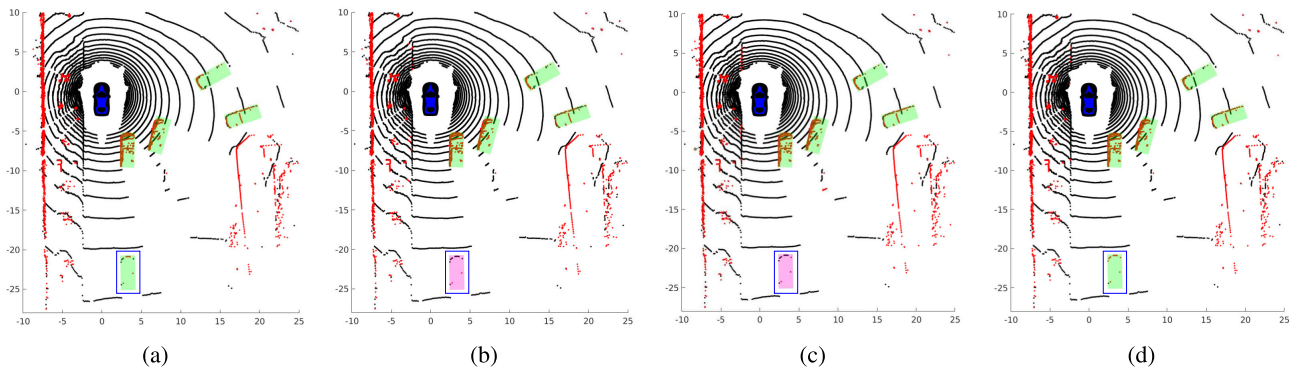


FIGURE 6. Common autonomous driving scene. Comparison between similar approaches. a) Initial classification. b) Initial classification and median filter. c) Algorithm in [1]. d) Proposed CB-MRF algorithm.

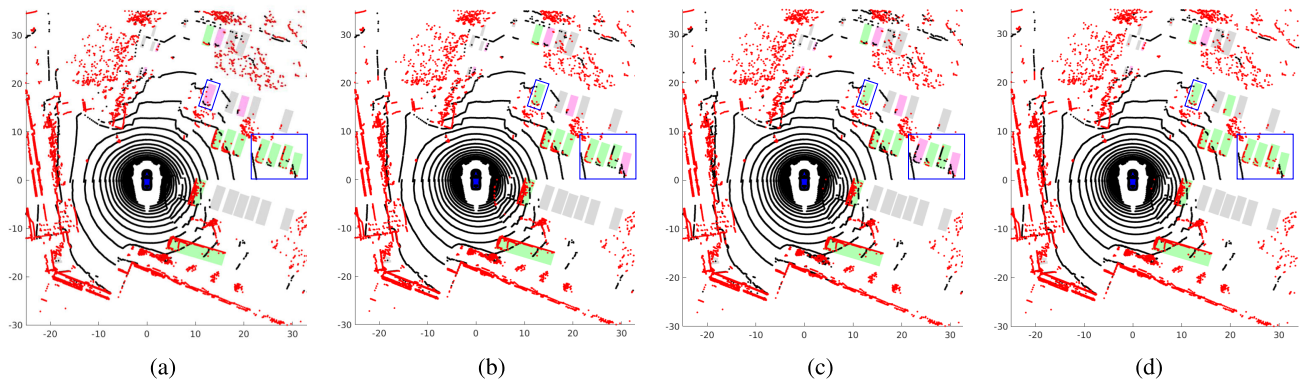


FIGURE 7. Complex autonomous driving scene. Comparison between similar approaches. a) Initial classification. b) Initial classification and median filter. c) Algorithm in [1]. d) Proposed CB-MRF algorithm.

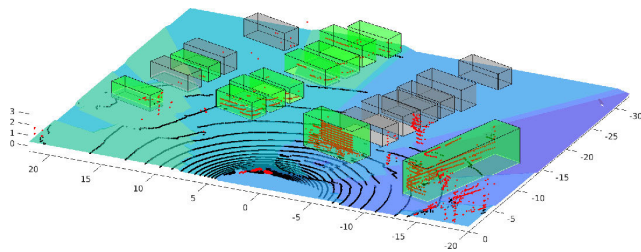


FIGURE 8. Classification and ground height grid map estimated by the proposed strategy.

vehicles, laser beams only hit the ground before the closer vehicle and under the farthest one (3D visualization is given in Fig. 8). Despite most of the points hit on the vehicles, no sharp features are captured, few sections of the vehicles can be detected as vertical structures and long distances can be found between the vehicles and previous ground points. Again, the CB-MRF is able to solve this situation because the previous channel-based algorithm estimates some of these points correctly. Conversely, the false-classifications and the lack of dense information wrongly guides the height grid map smoothed by the median filter. The approach of [1] also suffers in this situation since it relies in the identification of vertical structures inside the cells. Indeed, given a wrong initial reference and the lack of nearby ground points, the LBP confuses these obstacles with ground reliefs.

Fig. 8 shows the results of the proposed strategy from a 3D perspective. Only the right-side of the ego-vehicle is shown. The ground height grid map obtained from the MRF is also represented. As already introduced by [23], the explicit estimation of the ground surface provides valuable information for other autonomous driving tasks, such as path planning and control, taking into account height variations, pitch angle estimation, etc. The same colour scheme is used for the boxes and the point cloud. The height of the map is also coloured from the lowest height in dark blue to the highest in bright green. It can be seen how the estimated ground height is consistent with the point cloud and the ground truth 3D boxes. The worst case is found in the leftmost vehicle (the vehicle previously highlighted alone), for which the channel-based algorithm fails the classification of the majority of the points. Yet, as previously explained due to some lower points, the ground is estimated under the points of a higher layer hitting the vehicle.

D. QUANTITATIVE EVALUATION

A quantitative evaluation is conducted with the datasets from nuScenes. The subset *mini* is used since it provides four hundred samples from ten different representative scenes. In these samples, every point has a semantic label which is used as ground truth, and allows the following characterisation:

TABLE 2. Quantitative results from nuScenes dataset.

Approach	Precision	Recall	F-score	Balanced accuracy	Detected vehicles %	IoU
Channel-based	98.47	91.45	94.77	95.14	79.57	87.00
Channel-based with median filter	98.37	91.48	94.74	95.17	82.59	89.71
MRF-LBP based method in [1]	99.11	76.61	86.28	88.05	70.01	78.01
CSF [34]	94.59	91.90	92.98	93.67	86.72	90.03
CB-MRF	98.36	92.98	95.54	95.89	88.86	91.28

TABLE 3. Quantitative results from nuScenes dataset divided by radial distance only taking into account points relative to vehicles and drivable zone.

Approach	0-10m		10-20m		20-30m		30-40m		40-50m		50-60m	
	F-score	D.V. %	F-score	D.V. %	F-score	D.V. %	F-score	D.V. %	F-score	D.V. %	F-score	D.V. %
Channel-based	97.80	100	93.53	98.56	84.03	83.13	68.58	46.89	60.06	37.54	47.75	29.39
Channel-based with median filter	97.10	100	93.86	99.02	86.26	86.21	73.62	53.64	67.14	46.37	61.16	43.86
MRF-LBP based method in [1]	90.66	100	87.26	97.64	76.22	75.36	61.33	42.52	53.16	37.86	43.01	25.88
CSF [34]	92.67	100	87.79	99.08	81.60	88.61	78.81	66.89	74.99	61.20	73.23	54.83
CB-MRF	97.77	100	94.87	99.15	89.73	91.26	83.39	69.01	78.69	65.30	77.63	69.30

- True Positive (TP): Points classified as obstacle by the applied algorithm and as *non-flat* by the ground truth.
- False Positive (FP): Points classified as obstacle by the applied algorithm and as *flat* by the ground truth.
- True Negative (TN): Points classified as ground by the applied algorithm and as *flat* by the ground truth.
- False Negative (FN): Points classified as ground by the applied algorithm and as *non-flat* by the ground truth.

As the four analyzed algorithms share some common base-lines - e.g. the use of local geometric features - the results obtained with a different strategy is also included in the quantitative analysis. The selected algorithm is the CSF [34]. In this approach the point cloud is turned upside-down and a “simulated cloth” is used to cover it, obtaining an approximation of the ground surface. Then, points are classified with respect to this estimation. The implementation is taken from the library available on the *MathWorks* website [38] and parametrization is done taking as reference the works [33], [34] and adjusted with experimental tests: cloth hardness = 3, grid size = $1m$, height threshold = $0.2m$, iterations = 500, and time step = 0.65

Table 2 shows the results obtained for points in the range of 60 meters. Four classification metrics are included: Precision and Recall, in order to quantify the capability of correctly classifying and the probability of detecting obstacle points; F-score and Balanced Accuracy, which explain the overall performance of the algorithm. High scores are achieved by all the tested algorithms, proving that all of them are valid approaches. Yet, the proposed approach achieves slightly better scores. These results are useful to illustrate the general performance of the algorithm. Nevertheless, most of the points from the point cloud hit over the ground, facades or vegetation. These points are rather easy to classify and therefore high scores are also easily achieved. Besides, as it has been shown in the qualitative evaluation, vehicle detection is a challenging task, especially when few points hit on them. Given their reduced number of points, their influence on these metrics is low. For this reason, a complementary evaluation focusing on vehicles and the distance to the ego-vehicle is performed:

(i) **Score for vehicle detection:** A vehicle is counted as detectable if it has at least m laser hits and it is within the maximum analyzed distance. Likewise, it is counted as detected if at least m points are classified as obstacles. m is defined as 3 since it is the minimum number of points required to compute a convex-hull. Furthermore, the footprint indicates the size perceived of the vehicle, an important measurement for posterior tasks such as object segmentation or identification. For this analysis it is assumed that the best footprint achievable detection is defined by the convex-hull of all of its points. Knowing the points that truly belong to the vehicle, the quality of the vehicle detection is measured as the Intersection over Union (IoU) [39] between the convex hull obtained from all these points and the convex hull obtained from those points classified as obstacle. This metric is based on an uncertainty measurement defined by [40]. Since not all approaches detect the same vehicles, this score is computed with the vehicles detected by all approaches. The results of this complementary score show concordance with the qualitative analysis presented above. The proposed approach is able to detect more vehicles and with a better IoU than the other methods tested.

(ii) **Radial distance influence:** The point cloud distribution becomes sparser as the distance to the sensor increases. For nuScenes’ LiDAR sensor configuration a rough division would be:

- 0-20 meters: data can be defined as dense. Obstacles in this range are detected by multiple LiDAR points.
- 20-40 meters: data can be defined as relatively sparse. Relevant obstacles, such as vehicles, are detected by few points (see Fig. 6).
- 40 meters onwards: data can be defined as highly sparse. Few objects are detected in general.

In order to expose this behaviour, Table 3 shows the results obtained for different ranges of radial distance. Moreover, in order to highlight the results obtained for relevant data, only the points relative to vehicles and the drivable zone are considered. For ease of reading, only the F1-score and the vehicle detection score are given, as they are considered the most representative scores. The obtained results demonstrate how the obstacle-ground segmentation becomes

harder as distance increases. The effectiveness of all the tested methods decays with the increase of sparsity. Nevertheless, the CB-MRF algorithm is able to maintain an acceptable score, significantly higher than the other approaches in medium-long distances.

VI. CONCLUSION

This work presents the CB-MRF algorithm, a strategy for obstacle-ground classification in 3D LiDAR data specially focused on sloped terrain and sparse data. As main advantage, the proposed strategy combines the benefits of using channel-based algorithms, such as the analysis of different geometric features and the classification inheritance along points of the same the channel, with the ability of MRF-LBP based algorithms to model spatial dependencies and share information between nodes. In contrast to similar approaches that also perform a secondary step to improve the estimation, this work uses a higher order inference algorithm to correct the estimation, obtaining better results, particularly, in areas with low points density. With respect to similar approaches based on multi-label MRF and BP, this strategy uses an initial classification to model the data cost instead of geometric features inside the cells uniquely. This provides a more reliable description of the scene, reducing ambiguities between obstacles and ground.

Future research activities will be conducted on two different fronts: (i) real-time implementation, which is already in development with promising results using parallel programming with GPUs; and (ii) the exploration of temporal dependencies, which may lead to a more comprehensive description of the environment and thus a better estimation. Works such as [29] will be used as baseline.

REFERENCES

- [1] M. Zhang, D. D. Morris, and R. Fu, "Ground segmentation based on loopy belief propagation for sparse 3D point clouds," in *Proc. Int. Conf. 3D Vis.*, Oct. 2015, pp. 615–622.
- [2] J. Godoy, J. Pérez, E. Onieva, J. Villagrà, V. Milanés, and R. Haber, "A driverless vehicle demonstration on motorways and in urban environments," *Transport*, vol. 30, no. 3, pp. 253–263, Jan. 2015.
- [3] R. Dominguez, E. Onieva, J. Alonso, J. Villagrà, and C. Gonzalez, "LiDAR based perception solution for autonomous vehicles," in *Proc. 11th Int. Conf. Intell. Syst. Design Appl.*, Nov. 2011, pp. 790–795.
- [4] Y. Li and J. Ibanez-Guzman, "Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems," *IEEE Signal Process. Mag.*, vol. 37, no. 4, pp. 50–61, Jul. 2020.
- [5] C. Prenebida, O. Ludwig, and U. Nunes, "Exploiting LiDAR-based features on pedestrian detection in urban scenarios," in *Proc. 12th Int. IEEE Conf. Intell. Transp. Syst.*, Oct. 2009, pp. 1–6.
- [6] M. Saval-Calvo, L. Medina-Valdés, J. Castillo-Secilla, S. Cuenca-Asensi, A. Martínez-Álvarez, and J. Villagrà, "A review of the Bayesian occupancy filter," *Sensors*, vol. 17, no. 2, p. 344, Feb. 2017.
- [7] C. H. Rodriguez-Garavito, A. Ponz, F. Garcia, D. Martín, A. de la Escalera, and J. M. Armingol, "Automatic laser and camera extrinsic calibration for data fusion using road plane," in *Proc. 17th Int. Conf. Inf. Fusion (FUSION)*, Jul. 2014, pp. 1–6.
- [8] X. Hu, S. A. R. Florez, and A. Gepperth, "A multi-modal system for road detection and segmentation," in *Proc. IEEE Intell. Vehicles Symp.*, Dearborn, MI, United States, Jun. 2014, pp. 1365–1370.
- [9] D. Zermas, I. Izzat, and N. Papanikolopoulos, "Fast segmentation of 3D point clouds: A paradigm on LiDAR data for autonomous vehicle applications," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 5067–5073.
- [10] M. Himmelsbach and H.-J. Wuensche, "Tracking and classification of arbitrary objects with bottom-up/top-down detection," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2012, pp. 577–582.
- [11] M. Himmelsbach, F. V. Hundelshausen, and H.-J. Wuensche, "Fast segmentation of 3D point clouds for ground vehicles," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2010, pp. 560–565.
- [12] L. Chen, J. Yang, and H. Kong, "Lidar-histogram for fast road and obstacle detection," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 1343–1348.
- [13] J. Wu, J. Jiao, Q. Yang, Z.-J. Zha, and X. Chen, "Ground-aware point cloud semantic segmentation for autonomous driving," in *Proc. 27th ACM Int. Conf. Multimedia*, Oct. 2019, pp. 971–979.
- [14] J. Wu and Q. Yang, "Ground-distance segmentation of 3D LiDAR point cloud toward autonomous driving," *APSIPA Trans. Signal Inf. Process.*, vol. 9, pp. e24-1–e24-9, Nov. 2020. [Online]. Available: <https://www.cambridge.org/core/journals/apsipa-transactions-on-signal-and-information-processing/article/grounddistance-segmentation-of-3d-lidar-point-cloud-toward-autonomous-driving/91F72A086131E906889C5918E94B8152>
- [15] H. Badino, D. Huber, Y. Park, and T. Kanade, "Fast and accurate computation of surface normals from range images," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 3084–3091.
- [16] B. Li, "On enhancing ground surface detection from sparse lidar point cloud," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 4524–4529.
- [17] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, and G. Hoffmann, "Stanley: The robot that won the DARPA grand challenge," *J. Field Robot.*, vol. 23, no. 9, pp. 661–692, 2006.
- [18] J. Nitsch, J. Aguilar, J. Nieto, R. Siegwart, M. Schmidt, and C. Cadena, "3D ground point classification for automotive scenarios," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2603–2608.
- [19] P. Chu, S. Cho, S. Sim, K. Kwak, and K. Cho, "A fast ground segmentation method for 3D point cloud," *J. Inf. Process. Syst.*, vol. 13, no. 3, pp. 491–499, 2017.
- [20] P. M. Chu, S. Cho, J. Park, S. Fong, and K. Cho, "Enhanced ground segmentation method for lidar point clouds in human-centric autonomous robot systems," *Hum.-Centric Comput. Inf. Sci.*, vol. 9, no. 1, p. 17, Dec. 2019.
- [21] J. Cheng, Z. Xiang, T. Cao, and J. Liu, "Robust vehicle detection using 3D lidar under complex urban environment," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 691–696.
- [22] Z. Cheng, G. Ren, and Y. Zhang, "Ground segmentation algorithm based on 3D lidar point cloud," in *Proc. Int. Conf. Mech., Electr., Electron. Eng. Sci. (MEEES)*, 2018, pp. 16–21.
- [23] J. Rieken, R. Matthaei, and M. Maurer, "Benefits of using explicit ground-plane information for grid-based urban environment modeling," in *Proc. 18th Int. Conf. Inf. Fusion (Fusion)*, Jul. 2015, pp. 2049–2056.
- [24] A. Kampker, M. Sefati, A. S. Abdul Rachman, K. Kreisköther, and P. Campoy, "Towards multi-object detection and tracking in urban scenario under uncertainties," in *Proc. 4th Int. Conf. Vehicle Technol. Intell. Transp. Syst.*, 2018, pp. 1–12.
- [25] C. Guo, W. Sato, L. Han, S. Mita, and D. McAllester, "Graph-based 2D road representation of 3D point clouds for intelligent vehicles," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2011, pp. 715–721.
- [26] J. Byun, K. Na, B. Seo, and M. Roh, "Drivable road detection with 3D point clouds based on the MRF for intelligent vehicle," in *Field and Service Robotics* (Springer Tracts in Advanced Robotics), vol. 105, L. Mejias, P. Corke, and J. Roberts, Eds. Cham, Switzerland: Springer, 2015. [Online]. Available: https://link.springer.com/chapter/10.1007%2F978-3-319-07488-7_4#citeas, doi: 10.1007/978-3-319-07488-7_4.
- [27] W. Huang, H. Liang, L. Lin, Z. Wang, S. Wang, B. Yu, and R. Niu, "A fast point cloud ground segmentation approach based on coarse-to-fine Markov random field," *IEEE Trans. Intell. Transp. Syst.*, early access, Apr. 21, 2021, doi: 10.1109/TITS.2021.3073151.
- [28] T. Chen, B. Dai, R. Wang, and D. Liu, "Gaussian-process-based real-time ground segmentation for autonomous land vehicles," *J. Intell. Robot. Syst.*, vol. 76, nos. 3–4, pp. 563–582, Dec. 2014.
- [29] L. Rummelhard, A. Paigwar, A. Negre, and C. Laugier, "Ground estimation and point cloud segmentation using spatiotemporal conditional random field," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 1105–1110.

- [30] Y. Lyu, L. Bai, and X. Huang, "Real-time road segmentation using LiDAR data processing on an FPGA," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2018, pp. 1–5.
- [31] M. Velas, M. Spanel, M. Hradis, and A. Herout, "CNN for very fast ground segmentation in velodyne LiDAR data," in *Proc. IEEE Int. Conf. Auto. Robot Syst. Competitions (ICARSC)*, Apr. 2018, pp. 97–103.
- [32] W. L. Lu, K. P. Murphy, J. J. Little, A. Sheffer, and H. Fu, "A hybrid conditional random field for estimating the underlying ground surface from airborne LiDAR data," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 8, pp. 2913–2922, Aug. 2009.
- [33] T. Shi, D. Zhong, and L. Bi, "A new challenge: Detection of small-scale falling rocks on transportation roads in open-pit mines," *Sensors*, vol. 21, no. 10, p. 3548, May 2021.
- [34] W. Zhang, J. Qi, P. Wan, H. Wang, D. Xie, X. Wang, and G. Yan, "An easy-to-use airborne LiDAR data filtering method based on cloth simulation," *Remote Sens.*, vol. 8, no. 501, pp. 1–22, Jun. 2016.
- [35] J. H. Kappes, B. Andres, F. A. Hamprecht, C. Schnörr, S. Nowozin, D. Batra, S. Kim, B. X. Kausler, T. Kröger, J. Lellmann, N. Komodakis, B. Savchynskyy, and C. Rother, "A comparative study of modern inference techniques for structured discrete energy minimization problems," *Int. J. Comput. Vis.*, vol. 115, no. 2, pp. 155–184, Nov. 2015.
- [36] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient belief propagation for early vision," *Int. J. Comput. Vis.*, vol. 70, no. 1, pp. 41–54, Oct. 2006.
- [37] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. Erin Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "NuScenes: A multimodal dataset for autonomous driving," 2019, *arXiv:1903.11027*. [Online]. Available: <http://arxiv.org/abs/1903.11027>
- [38] W. Zhang, "Mathworks–CSF (cloth simulation filter)," RAMM Lab., School Geography, Beijing Normal Univ., Beijing, China, Tech. Rep., 2021. [Online]. Available: <http://ramm.bnu.edu.cn/>
- [39] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, Jan. 2014.
- [40] G. P. Meyer and N. Thakurdesai, "Learning an uncertainty-aware object detector for autonomous driving," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 10521–10527.



VÍCTOR JIMÉNEZ was born in Madrid, Spain, in 1993. He received the degree in industrial electronics and automation engineering from the Universidad Carlos III de Madrid, in 2017, and the master's degree in automation and robotics from the Universidad Politécnica de Madrid, in 2019. From 2018 to 2019, he received a research grant focused on mobile robots and fuzzy logic and competed in the SEAT Autonomous Driving Challenge 2018 as a member of AUTOPIA Program's Team. In 2020, he joined this program as a Predoctoral Researcher for the Centre for Automation and Robotics (CSIC-UPM). His research interests include autonomous driving, perception of the environment, localization, and information integrity.



JORGE GODOY was born in Maracay, Venezuela, in 1986. He received the degree in electronics engineering from Universidad Simón Bolívar, in 2008, and the M.E. and Ph.D. degrees in automation and robotics from the Universidad Politécnica de Madrid, in 2011 and 2013, respectively. From 2013 to 2017, he was a Technical Coordinator of the AUTOPIA Program funded by research contracts from national and European research projects. His research interests include intelligent transportation systems, autonomous driving, path planning, and embedded AI-based control for autonomous vehicles. In 2009, he was granted with a Predoctoral JAE Fellowship from CSIC for researching on autonomous vehicles at the Centre of Automation and Robotics (UPM-CSIC). In November 2017, he was granted with Juan de la Cierva Fellowship for postdoctoral research at the Universidad Politécnica de Madrid.



ANTONIO ARTUÑEDO received the B.Sc. degree in electrical engineering from the Universidad de Castilla-La Mancha, Spain, in 2011, the M.Sc. degree in industrial engineering from the Universidad Carlos III de Madrid, in 2014, and the Ph.D. degree in automation and robotics from the Technical University of Madrid (UPM), Spain, in 2019, with the AUTOPIA Program. During his predoctoral period, he made a research stay at the Integrated Vehicle Safety Group, TNO, The Netherlands, in 2017. He is currently a Postdoctoral Researcher with the Centre for Automation and Robotics (CSIC-UPM), AUTOPIA Group, Madrid, Spain. He has been working on both national and European research projects in the scope of autonomous vehicles. He has published and peer-reviewed multiple journals and conference papers focused in this field. His research interests include system modeling and simulation, intelligent control, motion planning, and decision-making systems. His thesis won the Prize to the Best Ph.D. on Intelligent Transportation Systems 2020 by the Spanish Chapter of the IEEE-ITS Society.



JORGE VILLAGRA graduated in industrial engineering from the Universidad Politécnica de Madrid, in 2002. He received the Ph.D. degree in real-time computer science, robotics and automatic control from the École des Mines de Paris, France, in 2006. From 2007 to 2009, he was a Visiting Professor with the University Carlos III de Madrid, Spain. From August 2013 to August 2016, he led the Department of ADAS and Highly Automated Driving Systems, Ixion Industry and Aerospace SL, where he also coordinated all the activities in the EU research and development funding programmes. He has been leading AUTOPIA Program at CSIC, since October 2016. He has developed his research activity in six different entities with a very intense activity in project setup and management, through over 40 international and national research and development projects, where he is or has been IP of 17 of these projects. He has published over 100 articles in international journals and in international conferences on autonomous driving, intelligent transportation systems, model-free control, and probabilistic approaches for embedded components in autonomous vehicles.

...