LiDAR Ground Segmentation with Gaussian Mixture Models

Felix Neumann[1], Frederik Deroo[1], Georg von Wichert[1], and Darius Burschka[2]

***Abstract*— Accurate ground segmentation is an important perception task, not only for estimating drivable surfaces, but also as a precursor for tasks such as object clustering or dynamic object segmentation and tracking in autonomous vehicles. Model-based methods have made substantial progress on this problem in recent times, but place their focus on generating a model of the ground, while modeling non-ground objects is not emphasized. However, modeling such objects can provide important additional information as evidence for non-ground points. We propose a Gaussian Mixture Model-based environment model to estimate the likelihood that local regions belong to the ground or non-ground objects, which can be queried at arbitrary positions in space. Additionally, we extend this approach to fuse information from multiple past sensor frames for more accurate ground estimation. We experimentally validate our approach on the SemanticKITTI dataset, where notably our single-frame configuration outperforms state-of-the-art multi-frame methods.**

I. Introduction

Autonomous transportation systems and autonomous mobile robots are gaining increasing relevance with advances in sensor technologies and processing capabilities. Such systems need to build an internal model of their surroundings to navigate safely and avoid collisions. Ground segmentation is a crucial task for building such models to, for example, determine drivable areas and as a precursor for object clustering [1], [2] and dynamic object detection and tracking [3].

While current approaches to ground segmentation have shown compelling performance, they are mostly focused on modeling the ground through local planes [4], [5] and elevation grids [6], [7]. These methods emphasize the model of the ground, but do not fully build models of non-ground objects, rather treating them as outlier points to the model of the ground. Points are then classified as ground if they lie within the boundaries of the ground model, which are often set as a fixed threshold, and as non-ground if they lie outside them. However, modeling both the ground and non-ground objects allows for a clear definition of segmentation boundaries between the two classes, and is therefore beneficial for accurate ground segmentation. Furthermore, heuristics-based methods often only supply a binary classification label and do not reason about uncertainty, which is crucial in safety-oriented systems. Finally, few approaches fully leverage the temporal persistence of the ground and only use single sensor frames, which can lead to inaccuracies in areas with sparse sensor measurements.

[1]Siemens Technology, Friedrich-Ludwig-Bauer-Straße 3, 85748 Garching, Germany, {neumann.felix, frederik.deroo, georg.wichert}@siemens.com

[2]Machine Vision and Perception Group, School of Computation, Information and Technology, Technical University of Munich, Friedrich-Ludwig-Bauer-Straße 3, 85748 Garching, Germany burschka@tum.de

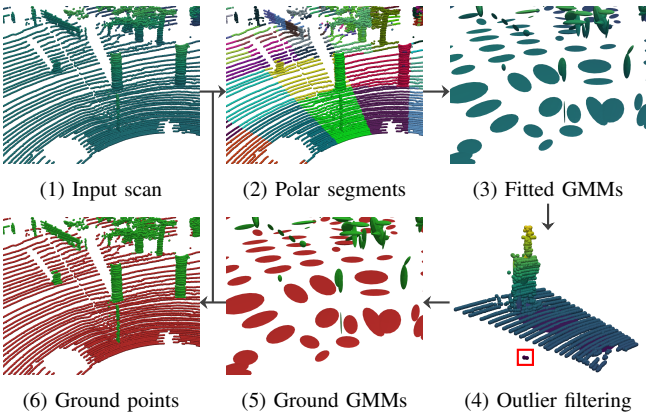


Fig. 1. Overview of our approach. An input scan (1) is first segmented according to the concentric zone model from Patchwork++ [4] (2). A 3D Gaussian Mixture Model (GMM) is fitted to the points in each segment (3) and used to identify outlier points below the lowest fitted Gaussian (4). The likelihood of each Gaussian to represent ground (red) and non-ground (green) is calculated using the three criteria described in Section III-C. The most likely class per Gaussian is visualized (5). Finally, the fitted model is queried with 3D points (for example the input scan) and a predicted ground state for each point is produced (6).

To address these issues, we propose the use of generative modeling in the form of Gaussian Mixture Models (GMMs) to estimate a continuous probability distribution that models the likelihood of a point in 3D space belonging to ground or non-ground objects. Specifically, our contributions are:

- We propose a probabilistic, parallelizable ground segmentation approach that is based on modeling range sensor point clouds as a collection of GMMs.
- Furthermore, we propose a multi-frame sensor fusion strategy that consistently improves performance over single-frame inference.
- We evaluate our approach on SemanticKITTI and demonstrate that it outperforms existing state-of-the-art (SotA) methods on the task of ground segmentation using both individual sensor frames and multiple sequential frames. Notably, our single scan configuration outperforms prior multi-frame methods.

II. Related Work

A. Gaussian Mixture Models for 3D Point Clouds

Recently, approximating 3D point cloud data using GMMs has gained interest in the domains of mapping and localization. However, the computational cost of fitting GMMs using classical approaches such as the Expectation-Maximization (EM) algorithm pose a challenge. Normal Distributions Transforms (NDTs) [8], [9] approximate GMMs by augmenting voxel-based representations with Gaussian

representations. A Gaussian distribution is fitted to measurement points falling into each voxel to build a 3D map, which can be used for localization of autonomous systems [10]. NDTs only approximate a full GMM, as fitting a GMM to large point clouds using EM is computationally expensive. Eckart *et al.* [11] address this issue by proposing a parallelized, hierarchical formulation of GMMs (HGMMs), which builds a tree of small GMMs using EM in real time by leveraging the computational power of modern Graphics Processing Units (GPUs). Their approach was later applied to real-time registration of LiDAR scans in outdoor scenarios [12]. Li *et al.* [13] propose a memory efficient approach to approximately fitting GMMs to depth images based on heuristics without iterative optimization and use this representation to iteratively build a GMM occupancy map [14]. We take inspiration from NDTs and Eckart *et al.* [11] in our approach by approximating LiDAR scans using a collection of small GMMs that are distributed across discrete segments of the environment, where we optimize the parameters of the GMMs in a parallelized manner on the GPU. This approach strikes a balance between NDTs and HGMMs. NDTs offer complete control on how Gaussians are distributed throughout 3D space and are very computationally efficient, but are limited to modeling the data as a single Gaussian per voxel. HGMMs, on the other hand, adapt to the data distribution, but are computationally more expensive and offer little control over the distribution of Gaussians in 3D space. Our approach takes the best of both worlds and offers control over how GMMs are distributed in space, while leveraging the adaptability of each GMM to local data. The computational complexity of our method is equivalent to a single hierarchy level in HGMMs, thus making it cheaper and faster to compute than HGMMs.

B. Model-based Ground Segmentation

Classical approaches to ground segmentation have been of interest for autonomous vehicles, with successful approaches being applied during the 2007 DARPA Urban Challenge. Himmelsbach *et al.* [15] propose an efficient line-fitting based approach to ground segmentation. Similar approaches have been used in applications such as object clustering [1] and dynamic object detection [3] to remove the ground as a precursor task. Shen *et al.* [16] coarsely classify ground points based on an elevation metric, followed by a smoothing of the class estimates in the range image of the sensor. Lim *et al.* [4] propose Patchwork, which segments the point cloud according to a concentric zone model in polar coordinates, estimates a local ground plane in each segment based on principal component analysis (PCA) of measurement points and classifies whether the plane represents the ground based on a set of thresholds. Lee *et al.* [5] extend this work to Patchwork++, which introduces robust filtering of outlier points and an adaptive estimation of the parameters used to classify ground points. Steinke *et al.* [7] propose to incrementally build an elevation map based on the assumption that the vertical distribution of points belonging to the ground is narrow in local grid cells. We take a similar approach to the works of Patchwork and Patchwork++ by splitting the point cloud into segments in polar coordinates, but instead of modeling only the ground, we model both ground and non-ground objects using GMMs in each segment. We furthermore exploit the temporal aspect of ground segmentation introduced by GroundGrid by proposing an efficient query method to fuse information from multiple sequential GMMs.

C. Learning-based Ground Segmentation

With the recent successes of deep learning in perception tasks, several learning-based approaches to ground segmentation have been introduced. Velas *et al.* [17] propose a lightweight convolutional neural network (CNN) that operates in the range view of a range sensor. Paigwar *et al.* [6] propose GndNet, which estimates the ground elevation in a 2D grid map based on an input point cloud using a 2D CNN, followed by a conditional random field. Points are classified based on the difference of elevation to the estimated elevation map. He *et al.* [18] employ a combination of CNNs on a polar discretization of the point cloud in Birds-Eye-View (BEV) and PointNet [19] to encode point clouds the BEV representation. Each polar sector is then classified as ground or non-ground and points are labeled based on the sector they fall into. While not specifically tuned to ground segmentation, LiDAR semantic segmentation methods such as 2DPass [20] can also be applied to the task of ground segmentation without retraining, by mapping the semantic classes they were trained on to the classes of ground and non-ground. We show that our model-based approach outperforms domain-specific deep learning methods.

III. Approach

Our approach consists of the following processing steps, which are also illustrated in Figure 1 and will be described in more detail in this section:

1) Fit a collection of GMMs to an input point cloud.
2) Identify outlier points based on the GMMs.
3) Estimate the likelihood of each Gaussian to represent ground or non-ground.
4) Calculate the ground probability of query points from the model.

This query based approach can be extended to fuse the information contained in multiple GMMs from sequential sensor measurements to improve the accuracy of the estimated ground points.

A. Problem Formulation

Given a 3D point cloud $\mathbf{S} := \{\mathbf{s}_1, ..., \mathbf{s}_N\}$ consisting of N points $\mathbf{s}_i \in \mathbb{R}^3$, we aim to estimate the probability that a point belongs to the ground or other non-ground objects for each query point. Thus, given a set of query points, our method produces an estimated probability vector $\hat{\mathbf{p}}_g := \{\hat{p}_{g,1}, ..., \hat{p}_{g,N}\}$ consisting of point probabilities $\hat{p}_{g,i} \in [0, 1]$. This probability can be used to determine the set of ground points $\mathbf{S}_g$ and non-ground points $\mathbf{S}_{\neg g}$ based on a threshold τ_g as $\mathbf{S}_g = \{\mathbf{s}_i \in \mathbf{S} | \hat{p}_{g,i} \geq \tau_\mathrm{g}\}$ and $\mathbf{S}_{\neg g} = \{\mathbf{s}_i \in \mathbf{S} | \hat{p}_{g,i} < \tau_\mathrm{g}\}$.

B. Gaussian Mixture Model Representation

Point clouds produced by range sensors suffer from issues such as noise and varying sparsity, which makes inferring abstract information about each point directly difficult. To address these issues, we approximate the distribution of points in a point cloud using a mixture of 3D Gaussians. This representation summarizes characteristics of local sections of the environment within each 3D Gaussian, which allows us to estimate the ground state of the segment based on the parameters of the fitted Gaussian.

The EM algorithm is commonly used to optimize the parameters of the Gaussian distributions in a GMM to approximate a set of data points. However, directly fitting a GMM with sufficient capacity to large point clouds in real time is intractable, since the complexity of EM for a GMM is $\mathcal{O}(NK)$ for a point cloud with N points and a GMM with K Gaussian components. While hierarchical GMMs have been proposed and have been shown to be real time capable when implemented on GPUs, they offer little control over the distribution of Gaussian components in the scene. This leads to areas with high point sparsity to be represented by few, large Gaussians, which are not representative enough to determine local features. Inspired by Patchwork [4] and Patchwork++ [5], we therefore first segment the point cloud in polar coordinates using the concentric zone model proposed by Lim *et al.* [4], resulting in B point subsets $\mathbf{S}_b$ where $\mathbf{S} = \cup_{b=1}^{B} \mathbf{S}_b$. We then fit a Gaussian mixture model with a maximum of $K_{\max}$ components to the points in each segment. Each component c of the GMM in segment b of the concentric zone model consists of a weight $\pi_{b,c}$, a mean $\boldsymbol{\mu}_{b,c} \in \mathbb{R}^3$ and a covariance $\boldsymbol{\Sigma}_{b,c} \in \mathbb{R}^{3\times3}$.

1) GMM Initialization: We find that using all $K_{\max}$ Gaussians in sparsely populated segments can sometimes lead to an oversegmentation of points, resulting in noisy Gaussians that are unsuited for ground estimation. Therefore, we initialize one Gaussian per $N_{\min}$ points in a segment of the concentric zone model, up to a maximum of $K_{\max}$ components. Thus, the number of initialized Gaussians K_b in a segment b is given by $K_b = \min(\lceil \frac{N_b}{N_{\min}} \rceil, K_{\max})$ for N_b points in the segment. The means of the K_b components are initialized with the mean of the points $\mathbf{S}_b$ in the segment in the horizontal plane (e.g., the x- and y-axes in common LiDAR coordinate systems) and are distributed linearly between the minimum and maximum heights of points in $\mathbf{S}_b$. The weight of each component c in the segment is initialized as $\pi_{b,c} = \frac{1}{K_b}$ and the covariance is set to a unit covariance matrix.

2) GMM Optimization: We employ Expectation Maximization to optimize the parameters of the initialized Gaussians. EM iteratively optimizes the parameters of the Gaussian components with alternating expectation and maximization steps. During the expectation step, the likelihood $\gamma_{b,c,i}$ between each measurement point $\mathbf{s}_{b,i}$ in a segment and each Gaussian component in that segment is given by:

$$\gamma_{b,c,i} = \frac{\pi_{b,c} p(\mathbf{s}_{b,i}|\pi_{b,c}, \boldsymbol{\mu}_{b,c}, \boldsymbol{\Sigma}_{b,c})}{\sum_{c=1}^{K_b} \pi_{b,c} p(\mathbf{s}_{b,i}|\pi_{b,c}, \boldsymbol{\mu}_{b,c}, \boldsymbol{\Sigma}_{b,c})},$$

where $p(\mathbf{s}_{b,i}|\pi_{b,c}, \boldsymbol{\mu}_{b,c}, \boldsymbol{\Sigma}_{b,c})$ is the probability density of the Gaussian component c at point $\mathbf{s}_{b,i}$. During the maximization step, the likelihood is maximized with respect to the parameters of the GMM components, which are updated in closed form as:

$$\begin{aligned}
\pi'_{b,c} &= \sum_i \frac{\gamma_{b,c,i}}{N_b} \\
\boldsymbol{\mu}'_{b,c} &= \frac{\sum_i \gamma_{b,c,i} \mathbf{s}_{b,i}}{\sum_i \gamma_{b,c,i}} \\
\boldsymbol{\Sigma}'_{b,c} &= \frac{\sum_i \gamma_{b,c,i} \mathbf{s}_{b,i} \mathbf{s}_{b,i}^\top}{\sum_i \gamma_{b,c,i}} - \boldsymbol{\mu}'_{b,c} \boldsymbol{\mu}'^\top_{b,c}.
\end{aligned}$$

At every iteration, we calculate how many points are assigned to each Gaussian component, i.e., for how many points a Gaussian is the component with the highest likelihood $\gamma_{b,c,i}$. Gaussians that are assigned fewer than τ_N points are removed to avoid oversegmentation.

C. Ground Likelihood of Gaussians

Once the GMMs of each segment are fitted to the point cloud, each component represents a local subset of points and approximates their distribution in space. We estimate the probability that a Gaussian represents ground or non-ground objects based on three assumptions related to the shape, orientation and elevation of the ground. For all three assumptions we employ a parameterizable likelihood function of the form:

$$p(x, \alpha, \beta) = 1 - \frac{1}{1 + e^{-\alpha(x-\beta)}}.$$

This is an inverted sigmoid function of input x with slope α and offset β. We choose this function as it allows us to map unbounded properties of the Gaussian components to representative probabilities that indicate evidence for the ground and non-ground classes. Furthermore, we can define an uncertain transition region in the function and high confidence regions that are robust to outliers where the function saturates based on the manually defined parameters α and β. To choose suitable values, we recommend to set β to a threshold below which the components in the GMM tend to represent ground segments. Then, α should be tuned such that $p(x, \alpha, \beta)$ evaluates to a sufficiently high value at $x = 0$. We use $p(0, \alpha, \beta) > 0.9$ in our parameter tuning. A higher value for β will lead to higher estimated likelihoods and thus a higher recall and possibly lower precision.

1) Ground Shape: Generally, we assume local sections of ground to be smooth and flat. This should be reflected by a Gaussian that represents a local set of ground points, where the variance of the Gaussian is low in at least one direction. We calculate the likelihood that a Gaussian is part of the ground based on its flatness as $p_{g,\lambda} = p(\lambda_{\min}, \alpha_\lambda, \beta_\lambda)$, where $\lambda_{\min}$ is the smallest eigenvalue of the covariance matrix. This causes a high ground probability for Gaussians that are sufficiently flat in at least one direction.

2) Ground Orientation: The second assumption we make about ground components is that they are oriented close to parallel to a horizontal plane. To assess this, we estimate the ground probability based on the angle $\theta = \arccos(\mathbf{v}_{\min} \cdot \mathbf{n}_z)$ between the eigenvector $\mathbf{v}_{\min}$ corresponding to $\lambda_{\min}$ and the vertical axis $\mathbf{n}_z$ (e.g., the z axis in common LiDAR coordinates) as $p_{g,\theta} = p(\theta, \alpha_\theta, \beta_\theta)$. This assigns a high ground probability to Gaussians whose larger principal components are close to parallel to the horizontal plane in sensor coordinates. While this constraint could hinder the detection of upwards sloping ground, we find that in practice we can set this parameter large enough to cover gradients up to 45 degrees without compromising on accuracy.

3) Ground Elevation: Unlike Patchwork [4] and Patchwork++ [5], we make no assumption about the mounting height of the sensor, but assume that ground sections are located near the measurement points with the smallest elevation in their respective segment. Therefore, we assume that the means of Gaussians that represent ground sections have an elevation that is close to the lowest points within a segment b. We calculate this elevation distance as $e = \boldsymbol{\mu}_{b,c} \cdot \mathbf{n}_z - \min(\mathbf{S}_{b,e})$, where $\boldsymbol{\mu}_{b,c} \cdot \mathbf{n}_z$ is the vertical elevation of the mean of the Gaussian and $\mathbf{S}_{b,e}$ is the set of elevations of all points in a segment. We calculate the ground likelihood based on the elevation as $p_{g,e} = p(e, \alpha_e, \beta_e)$, which yields a high probability for Gaussian means close to the minimum points and a low probability for Gaussians with large elevations above the minimum measurement points.

The three likelihoods $p_{g,\lambda}$, $p_{g,\theta}$ and $p_{g,e}$ serve as indicators whether a Gaussian likely represents the ground, and are used to determine the overall likelihood of a query point belonging to the ground, as described in Section III-E.

D. Outlier Removal using GMMs

One issue with using the above criteria with real point cloud data are LiDAR beams that reflect off of multiple objects. This tends to happen when reflective objects, such as cars, are close to the LiDAR where the beam intensity is high, which results in perceived points that lie beneath the ground. This causes a low ground probability w.r.t elevation $p_{g,e}$, which results in false negative point classifications. To address this issue, we use the fitted Gaussian components to filter outlier points in a segment before calculating the minimum elevation of points in that segment.

As there are only few outlier points generally, and we remove Gaussians supported by fewer than τ_N points, outliers should not be represented well by the GMM. Therefore, we classify points where $\mathbf{s}_{b,e} - \min_(\boldsymbol{\mu}_{b,e}) < \tau_{\text{o}}$ as outlier points and exclude them from the point elevation set $\mathbf{S}_{b,e}$ to assess the ground probability with respect to elevation. However, in proximity to the sensor, it is possible that enough outlier points exist to be assigned a Gaussian component. We filter these out by adaptively estimating the sensor height above the ground and not considering Gaussians that lie well below the sensor height in polar segments close to the sensor. For this, we maintain a moving average of the elevation of Gaussians with high ground likelihood in each segment in the first zone of the concentric zone model [4] over consecutive sensor frames. We calculate the mean $\bar{z}_s$ and standard deviation $\sigma_{z,s}$ of the moving averages in each segment and exclude Gaussians that lie below $\bar{z}_s - 3\sigma_{z,s}$ from the Gaussians considered during outlier point removal in polar segments in the first zone. We choose three standard deviations, because the expected fraction of elevation samples that lies outside this range is less than one percent. This minimizes the risk of incorrectly filtering an actual ground Gaussian from the calculation of the minimum ground elevation.

E. Gaussian Mixture Model Query

Once the GMMs in each segment are fitted to the point cloud data and the ground probabilities are calculated according to the likelihood functions described in Section III-C, the model represents a continuous probability distribution over the measurements and the ground likelihood in each segment. This model can be queried with a set of 3D query points $\mathbf{S}_q$ to yield a ground likelihood of each point. To estimate the class of a query point, we identify which segment of the concentric zone model it falls into to select the respective GMM. Next, the association likelihood $\gamma_{b,c,i}$ between a query point $\mathbf{s}_{q,b,i}$ and the Gaussian components in a segment are calculated. For readability, we omit the segment index b for the notation of the following calculation. These association likelihoods are then used to form weighted sums of the three ground likelihoods of each Gaussian for the point as $p_{g,\lambda,i} = \sum_c \gamma_{c,i} p_{g,\lambda,c}$, $p_{g,\theta,i} = \sum_c \gamma_{c,i} p_{g,\theta,c}$, and $p_{g,e,i} = \sum_c \gamma_{c,i} p_{g,e,c}$. The overall predicted probability that the query point belongs to the ground is then given by the product of these three likelihoods as $\hat{p}_{g,i} = p_{g,\lambda,i} p_{g,\theta,i} p_{g,e,i}$.

F. Multi Frame Query

Since the information about the environment is limited within a single scan of the sensor, we propose an efficient multi-frame query method that can be used to estimate the ground likelihood for a point based on the GMMs from multiple frames. However, for this we additionally require the pose $T^f \in SE(3)$ of the sensor in world coordinates at which sensor frame f was observed. We maintain a buffer of the last N_f GMMs, of the ground likelihoods $p^f_{g,\lambda,c}$, $p^f_{g,\theta,c}$, and $p^f_{g,e,c}$ for each Gaussian component within them, and of the poses associated with the GMMs. Given a set of query points $\mathbf{S}^f_q$ relative to the sensor pose T^f_q, we query all available GMMs in the stored buffer. To this end, we begin by projecting the query points into the sensor frames of the stored GMMs using the buffered poses and the query pose. After projecting the points, we identify the segment of the concentric zone model of the past frames that each point falls into. The likelihood of each point is then estimated from the Gaussians of all segments that the respective query point falls into in each of the past frames. Thus, the association likelihood $\gamma_{b,c,i}$ is calculated not only for the Gaussians of a single segment b but for the segments that the point falls

into in each buffered frame f, thus forming:

$$\gamma^f_{b,c,i} = \frac{\pi^f_{b,c} p(\mathbf{s}^f_{q,b,i} | \pi^f_{b,c}, \boldsymbol{\mu}^f_{b,c}, \boldsymbol{\Sigma}^f_{b,c})}{\sum_{f=1}^{N_f} \sum_{c=1}^{K_b} \pi^f_{b,c} p(\mathbf{s}^f_{q,b,i} | \pi^f_{b,c}, \boldsymbol{\mu}^f_{b,c}, \boldsymbol{\Sigma}^f_{b,c})}.$$

Omitting the segment index b again in the following, the three ground likelihoods are then calculated analogously as $p_{g,\lambda,i} = \sum_f \sum_c \gamma^f_{c,i} p^f_{g,\lambda,c}$ for the ground likelihood w.r.t flatness, $p_{g,\theta,i} = \sum_f \sum_c \gamma^f_{c,i} p^f_{g,\theta,c}$ for the likelihood w.r.t. the ground orientation, and $p_{g,e,i} = \sum_f \sum_c \gamma^f_{c,i} p^f_{g,e,c}$ for the likelihood w.r.t elevation. The final point likelihood $\hat{p}_{g,i}$ is calculated as described in Section III-E.

IV. Experiments

In this section, we outline the evaluation protocol for our proposed approach and analyze its performance compared to other SotA ground segmentation approaches.

A. Evaluation Setting

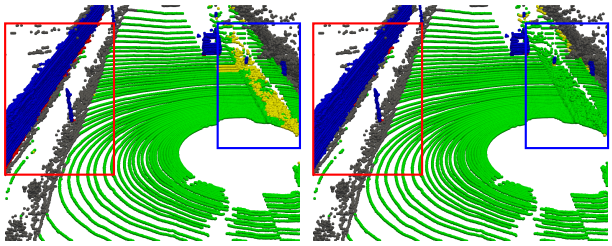
We use the SemanticKITTI dataset [21] to evaluate our approach in an urban autonomous driving context. To generate the ground truth ground labels, we follow the evaluation methodology of Lee *et al.* [5] and Steinke *et al.* [7] to label points belonging to the *road*, *sidewalk*, *lane marking*, *parking*, *terrain*, and *other ground* as ground points. The labels *unlabeled*, *outlier* and *vegetation* are ignored during the evaluation, as the *vegetation* class contains a large variety of objects that cannot conclusively be assigned to ground or non-ground objects. All remaining labels are considered as non-ground objects. For multi-frame ground segmentation, we use sensor poses estimated by KISS-ICP [22]. We evaluate our approach against SotA learning- and model-based approaches using the precision, recall, Intersection over Union (IoU), F1 score, and accuracy metrics.

B. Implementation Details

We implement our approach on the GPU using Pytorch [23] and Triton [24]. We set the maximum number of Gaussians per segment as $K_{\max} = 8$ where we initialize one Gaussian per $N_{\min} = 20$ points in a segment. The threshold for removing a Gaussian with insufficient support is set to $\tau_N = 10$. The parameters of the likelihood functions are empirically set to $\alpha_\lambda = 40$, $\beta_\lambda = 0.06$, $\alpha_\theta = \alpha_e = 4$, and $\beta_\theta = \beta_e = 0.8$. The outlier threshold is chosen to be $\tau_\text{o} = 0.5$, and the threshold for classifying ground points based on the estimated ground probability is set to $\tau_\text{g} = 0.5$. We determine these values empirically on the validation set of SemanticKITTI (Seq. 08).

C. Comparison on SemanticKITTI

We compare our approach to GndNet [6], 2DPass [20], Patchwork++ [5] and Groundgrid [7], as they respectively represent a learning-based ground segmentation approach, a SotA LiDAR semantic segmentation approach, a single-frame model-based approach and a multi-frame model-based approach to ground segmentation. We report the quantitative results in Table I, where we show that our approach outperforms the current SotA in terms of precision, F1 score, accuracy and IoU, while reaching a nearly identical recall. While we report the performance on the entire publically available dataset, it should be noted that the deep-learning approaches GndNet and 2DPass were trained on some of the sequences. Therefore, we mark results that contain training samples with parentheses where the dataset split is known and exclude these values from the assessment of best and second best performance. The results show that our single frame configuration not only outperforms the learning-based and model-based single scan approaches of GndNet and Patchwork++ by a significant margin, but also achieves higher performance than the multi-frame approach taken by GroundGrid, albeit at a small difference. Extending our approach to multiple frames consistently improves every metric across all sequences, and achieves the best performance on every sequence in terms of F1 score, Accuracy and IoU.



(a) Patchwork++ (b) Ours (single frame)

Fig. 2. Comparison between our approach and Patchwork++. True positive ground detections are shown in green, true negatives in blue, false positives in red, and false negatives in yellow. Our approach successfully classifies the uneven ground next to the road, while Patchwork++ produces false negative detections, including misclassifying an entire patch of ground on the road behind the car (blue box). Our approach also produces fewer false positives at the bottom of the wall on the left (red box).

Overall, our approach achieves higher precision than others, while reaching a similar recall. These metrics support our claim that modeling non-ground objects is beneficial to the task of ground segmentation, as our method produces fewer false positive estimates, where object points are classified as ground points. The high recall of GndNet likely originates from some evaluation sequences being included in the training data of the approach, but the exact train-validation split was not reported by the authors. Compared to 2DPass, our method achieves a higher recall and slightly lower precision on the validation set of 2DPass (Seq. 08). Even though the deep learning methods were evaluated on their training sequences, our multi-frame approach outperforms them in almost every metric, without requiring training and while being much more computationally efficient. Our approach overall offers the best balance between precision and recall, as shown by the IoU, F1 score and accuracy metrics. A qualitative comparison between our single-frame approach and Patchwork++ [5] is shown in Figure 2, where our method is more robust to uneven ground, and more accurate at the transition from objects to the ground.

We demonstrate the effectiveness of our multi-frame query approach in Table II, where we show that including additional past frames to estimate the ground state of query points consistently improves the F1 score, accuracy and IoU

TABLE I

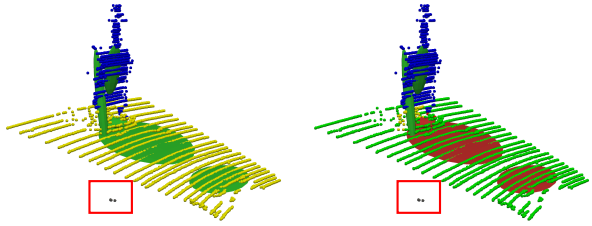
COMPARISON OF OUR PROPOSED APPROACH TO SOTA GROUND SEGMENTATION METHODS ON SEQUENCE 00-10 OF THE SEMANTICKITTI DATASET. BEST PERFORMANCE IS HIGHLIGHTED IN BOLD, SECOND BEST PERFORMANCE IS UNDERLINED. RESULTS OF DEEP-LEARNING METHODS ON KNOWN TRAINING DATA ARE PROVIDED IN PARENTHESES AND NOT CONSIDERED FOR BEST PERFORMANCE.

Method	Multi-frame	Seq 00	Seq 01	Seq 02	Seq 03	Seq 04	Seq 05	Seq 06	Seq 07	Seq 08	Seq 09	Seq 10	Mean
Precision													
GndNet	✗	92.40	96.54	93.74	95.60	97.30	89.58	96.15	90.09	95.09	93.81	88.34	93.51
2DPass	✗	(99.25)	(98.92)	(99.30)	(99.59)	(99.51)	(98.78)	(99.15)	(99.09)	**99.44**	(98.97)	(98.54)	(99.14)
Patchwork++	✗	94.99	**98.27**	95.96	96.81	98.18	92.65	97.86	93.29	97.03	96.06	92.81	95.81
GroundGrid	✓	96.05	98.01	97.36	97.96	99.08	95.19	97.82	95.31	97.25	97.25	95.38	96.97
Ours	✗	97.82	96.57	98.68	99.29	99.29	97.04	98.44	97.52	98.47	98.19	96.99	98.03
Ours	✓	**98.61**	97.34	**99.02**	**99.56**	**99.48**	**98.00**	**98.79**	**98.41**	99.00	**98.60**	**97.98**	**98.61**
Recall													
GndNet	✗	**99.50**	96.91	96.94	96.68	**99.06**	**98.69**	**99.00**	**99.44**	**98.74**	96.14	93.60	**97.70**
2DPass	✗	(98.13)	(96.84)	(97.48)	(97.60)	(97.40)	(96.92)	(98.23)	(97.53)	95.38	(96.10)	(95.47)	(97.01)
Patchwork++	✗	98.67	96.52	97.20	98.17	97.21	98.13	97.39	98.42	97.35	96.45	95.93	97.40
GroundGrid	✓	98.70	96.17	**97.71**	97.95	97.85	98.13	98.38	98.72	97.79	**96.91**	95.90	97.66
Ours	✗	97.44	96.18	96.31	98.15	98.22	96.86	97.67	97.26	96.95	96.04	95.02	96.92
Ours	✓	98.15	**98.05**	97.00	**98.80**	98.63	97.49	98.02	98.04	97.62	96.69	**96.09**	97.69
F1 score													
GndNet	✗	95.82	96.72	95.31	96.14	98.17	93.91	97.55	94.53	96.88	94.96	90.89	95.53
2DPass	✗	(98.69)	(97.86)	(98.38)	(98.59)	(98.45)	(97.84)	(98.69)	(98.30)	97.37	(97.51)	(96.98)	(98.06)
Patchwork++	✗	96.80	97.39	96.58	97.49	97.69	95.31	97.63	95.79	97.19	96.25	94.35	96.59
GroundGrid	✓	97.35	97.08	97.54	97.96	98.46	96.64	98.10	96.99	97.64	97.08	95.64	97.32
Ours	✗	97.63	96.38	97.48	98.71	98.75	96.95	98.05	97.39	97.70	97.10	95.99	97.47
Ours	✓	**98.38**	**97.69**	**98.00**	**99.18**	**99.05**	**97.74**	**98.40**	**98.23**	**98.31**	**97.64**	**97.03**	**98.15**
Accuracy													
GndNet	✗	95.53	94.88	93.20	93.99	97.10	93.10	96.46	94.52	95.91	93.08	89.81	94.33
2DPass	✗	(98.66)	(96.71)	(97.71)	(97.84)	(97.58)	(97.69)	(98.14)	(98.40)	96.69	(96.67)	(96.77)	(97.53)
Patchwork++	✗	96.64	95.96	95.08	96.08	96.39	94.79	96.63	95.88	96.38	94.90	93.75	95.68
GroundGrid	✓	97.24	95.50	96.48	96.84	97.60	96.32	97.29	97.08	96.97	96.05	95.25	96.60
Ours	✗	97.56	94.38	96.44	98.02	98.05	96.71	97.23	97.52	97.07	96.11	95.69	96.80
Ours	✓	**98.34**	**96.39**	**97.17**	**98.74**	**98.52**	**97.57**	**97.73**	**98.31**	**97.84**	**96.82**	**96.80**	**97.66**
IoU													
GndNet	✗	91.97	93.65	91.04	92.55	96.41	88.52	95.22	89.63	93.94	90.41	83.30	91.51
2DPass	✗	(97.41)	(95.82)	(96.81)	(97.22)	(96.94)	(95.77)	(97.42)	(96.66)	94.87	(95.15)	(94.14)	(96.20)
Patchwork++	✗	93.79	94.90	93.38	95.09	95.49	91.04	95.36	91.91	94.53	92.78	89.30	93.42
GroundGrid	✓	94.84	94.33	95.19	96.00	96.97	93.49	96.27	94.15	95.40	94.33	91.64	94.78
Ours	✗	95.37	93.01	95.08	97.46	97.53	94.07	96.17	94.92	95.51	94.37	92.30	95.07
Ours	✓	**96.81**	**95.49**	**96.08**	**98.37**	**98.12**	**95.58**	**96.85**	**96.51**	**96.67**	**95.38**	**94.23**	**96.37**

TABLE II

ABLATION ON THE EFFECT OF THE NUMBER OF PAST QUERY FRAMES ON PERFORMANCE AND QUERY TIME PER SCAN ON SEMANTICKITTI.

Number of frames	1	2	4	8	16
Mean F1 score	97.47	97.77	97.96	98.08	98.15
Mean Accuracy	96.80	97.18	97.42	97.57	97.66
Mean IoU	95.07	95.65	96.00	96.24	96.37
Mean query time (ms)	0.27	0.33	0.48	1.02	2.25



(a) Without outlier filtering (b) With outlier filtering

Fig. 3. Effect of our proposed outlier filtering. Red Gaussians are estimated to represent ground, green to represent non-ground. Green points are correctly classified ground, blue points correctly classified non-ground, and yellow points are incorrectly classified as non-ground. Without outlier filtering, the grey outlier points (red box) cause the ground likelihood w.r.t. elevation to be low for Gaussians representing the ground, thus causing actual ground points to be misclassified.

metrics, until saturation is reached at around 16 frames. Furthermore, we show that this additional performance comes at a small computational cost. Since we merely need to store the already generated GMMs of past LiDAR frames, only the runtime of the query function is affected by including additional frames. Due to our highly parallelized implementation on the GPU, querying 16 frames with approximately 120.000 query points only needs two additional milliseconds compared to the single frame query. We can even achieve nearly the same performance at the cost of only 750 additional microseconds if we reduce the number of query frames to eight. Overall, our method requires 4.6 ms to segment a point cloud into the concentric zone model, fit the GMMs and estimate the ground likelihood of Gaussians, in addition to the query times outlined above. Thus, the approach can run at 146-205 Hz depending on the number of query frames and is well suited to real-time applications. All inference times were measured on an Nvidia RTX A3000M laptop GPU.

We furthermore conduct ablation studies on the effect of

TABLE III

ABLATION ON THE EFFECT OF OUTLIER FILTERING, NUMBER OF GAUSSIANS PER GMM AND MULTI-FRAME QUERIES ON PERFORMANCE.

Outlier Filter	Multi-Frame	Num. Gauss.	Precision	Recall	F1 score
✗	✗	8	98.13	95.85	96.99
✓	✗	8	98.03	96.92	97.47
✗	✓	8	98.71	97.17	97.93
✓	✓	8	98.61	97.69	98.15
✓	✓	4	98.54	97.35	97.94
✓	✓	16	98.62	97.63	98.12

outlier filtering and the number of Gaussian components in a GMM in Table III. Outlier filtering slightly reduces the precision of our approach, but provides a large improvement to recall, and overall leads to better performance. The cause of this is visualized in Figure 3, where a few outlier points cause the likelihood w.r.t. elevation of the actual ground Gaussians to be low, resulting in many false negative estimates. We note that the multi-frame query approach is able to compensate for some outliers, shown by a smaller relative improvement in recall and F1 score due to the outlier filtering than in the single frame setting. This is reasonable, as outliers tend be inconsistently over time. We also show that using fewer than our chosen eight Gaussian components in each GMM reduces the performance of our approach, while using more Gaussians does not provide more accurate estimates and requires significantly more computation.

V. CONCLUSION

This paper presented a novel, probabilistic approach to ground segmentation based on generative modeling of 3D point clouds with 3D Gaussian Mixture Models. Experimental results show that the proposed approach outperforms current SotA ground segmentation models using both single sensor scans and sequential sensor data. Furthermore, the GMM-based representation is shown to be effective in suppressing adverse effects from outlier measurements. The highly parallelizable nature of the approach allows for an efficient implementation on GPUs for online execution in autonomous vehicles. While our approach offers strong performance in automotive scenarios, it makes the assumption that the sensor is oriented nearly parallel to the ground for the likelihood calculation and the outlier filtering steps. Furthermore, the multi-frame inference of our approach requires accurate sensor pose estimations to project points into past frames. To address these shortcomings, we believe that a tighter integration of the GMM representation with sensor pose estimation is promising future work to improve the reliability and robustness of ground segmentation.

REFERENCES

[1] I. Bogoslavskyi and C. Stachniss, "Fast range image-based segmentation of sparse 3D laser scans for online operation," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2016, pp. 163–169.

[2] F. Hasecke, L. Hahn, and A. Kummert, "FLIC: Fast Lidar Image Clustering," *arXiv preprint arXiv:2003.00575*, 2020.

[3] A. Reich and H.-J. Wuensche, "Detection of Moving Objects Based on Efficient Particle Tracking in 1.5D LiDAR Range Images," in *IEEE 26th Int. Conf. on Intelligent Transportation Systems (ITSC)*, 2023, pp. 1890–1895.

[4] H. Lim, M. Oh, and H. Myung, "Patchwork: Concentric zone-based region-wise ground segmentation with ground likelihood estimation using a 3D LiDAR sensor," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6458–6465, 2021.

[5] S. Lee, H. Lim, and H. Myung, "Patchwork++: Fast and robust ground segmentation solving partial under-segmentation using 3D point cloud," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2022, pp. 13 276–13 283.

[6] A. Paigwar, Ö. Erkent, D. Sierra-Gonzalez, and C. Laugier, "GndNet: Fast ground plane estimation and point cloud segmentation for autonomous vehicles," in *IEEE/RSJ Int. Conf. on intelligent robots and systems (IROS)*, 2020, pp. 2150–2156.

[7] N. Steinke, D. Goehring, and R. Rojas, "GroundGrid: LiDAR Point Cloud Ground Segmentation and Terrain Estimation," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 420–426, 2024.

[8] J. Saarinen, H. Andreasson, T. Stoyanov, J. Ala-Luhtala, and A. J. Lilienthal, "Normal Distributions Transform Occupancy Maps: Application to large-scale online 3D mapping," in *IEEE Int. Conf. on robotics and automation*, 2013, pp. 2233–2238.

[9] J. Saarinen, T. Stoyanov, H. Andreasson, and A. J. Lilienthal, "Fast 3D mapping in highly dynamic environments using normal distributions transform occupancy maps," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2013, pp. 4694–4701.

[10] J. Saarinen, H. Andreasson, T. Stoyanov, and A. J. Lilienthal, "Normal distributions transform Monte-Carlo localization (NDT-MCL)," in *IEEE/RSJ Int. Conf. on intelligent robots and systems*, 2013, pp. 382–389.

[11] B. Eckart, K. Kim, A. Troccoli, A. Kelly, and J. Kautz, "Accelerated Generative Models for 3D Point Cloud Data," in *Proc. of the IEEE Conf. on computer vision and pattern recognition*, 2016, pp. 5497–5505.

[12] B. Eckart, K. Kim, and J. Kautz, "HGMR: Hierarchical Gaussian Mixtures for Adaptive 3D Registration," in *Proc. of the European Conf. on computer vision (ECCV)*, 2018, pp. 705–721.

[13] P. Z.-X. Li, S. Karaman, and V. Sze, "Memory-Efficient Gaussian Fitting for Depth Images in Real Time," in *Int. Conf. on Robotics and Automation (ICRA)*, 2022, pp. 8003–8009.

[14] P. Li, S. Karaman, and V. Sze, "GMMap: Memory-Efficient Continuous Occupancy Map Using Gaussian Mixture Model," *IEEE Transactions on Robotics*, vol. 40, pp. 1339–1355, 2024.

[15] M. Himmelsbach, F. V. Hundelshausen, and H.-J. Wuensche, "Fast segmentation of 3D point clouds for ground vehicles," in *IEEE Intelligent Vehicles Symposium*, 2010, pp. 560–565.

[16] Z. Shen, H. Liang, L. Lin, Z. Wang, W. Huang, and J. Yu, "Fast Ground Segmentation for 3D LiDAR Point Cloud Based on Jump-Convolution-Process," *Remote Sensing*, vol. 13, no. 16, p. 3239, 2021.

[17] M. Velas, M. Spanel, M. Hradis, and A. Herout, "CNN for Very Fast Ground Segmentation in Velodyne LiDAR Data," in *IEEE Int. Conf. on Autonomous Robot Systems and Competitions (ICARSC)*, 2018, pp. 97–103.

[18] D. He, F. Abid, Y.-M. Kim, and J.-H. Kim, "SectorGSnet: Sector Learning for Efficient Ground Segmentation of Outdoor LiDAR Point Clouds," *IEEE Access*, vol. 10, pp. 11 938–11 946, 2022.

[19] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," in *Proc. of the IEEE Conf. on computer vision and pattern recognition*, 2017, pp. 652–660.

[20] X. Yan, J. Gao, C. Zheng, C. Zheng, R. Zhang, S. Cui, and Z. Li, "2DPASS: 2D Priors Assisted Semantic Segmentation on LiDAR Point Clouds," in *European Conf. on computer vision*. Springer, 2022, pp. 677–695.

[21] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "Semantickitti: A dataset for semantic scene understanding of lidar sequences," in *Proc. of the IEEE/CVF Int. Conf. on computer vision*, 2019, pp. 9297–9307.

[22] I. Vizzo, T. Guadagnino, B. Mersch, L. Wiesmann, J. Behley, and C. Stachniss, "KISS-ICP: In Defense of Point-to-Point ICP – Simple, Accurate, and Robust Registration If Done the Right Way," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 1029–1036, 2023.

[23] A. Paszke, "Pytorch: An imperative style, high-performance deep learning library," *arXiv preprint arXiv:1912.01703*, 2019.

[24] P. Tillet, H.-T. Kung, and D. Cox, "Triton: an intermediate language and compiler for tiled neural network computations," in *Proc. of the 3rd ACM SIGPLAN Int. Workshop on Machine Learning and Programming Languages*, 2019, pp. 10–19.