Motion-based Detection and Tracking in 3D LiDAR Scans

Ayush Dewan  Tim Caselitz  Gian Diego Tipaldi  Wolfram Burgard

Abstract—Robots are expected to operate autonomously in increasingly complex scenarios such as crowded streets or heavy traffic situations. Perceiving the dynamics of moving objects in the environment is crucial for safe and smart navigation and therefore a key enabler for autonomous driving. In this paper we present a novel model-free approach for detecting and tracking dynamic objects in 3D LiDAR scans obtained by a moving sensor. Our method only relies on motion cues and does not require any prior information about the objects. We sequentially detect multiple motions in the scene and segment objects using a Bayesian approach. For robustly tracking objects, we utilize their estimated motion models. We present extensive quantitative results based on publicly available datasets and show that our approach outperforms the state of the art.

I. INTRODUCTION

One of the major goals in the area of mobile robotics is to develop robot systems that can robustly navigate to accomplish different tasks such as surveillance and transportation. As the environment in which a robot is expected to operate typically cannot be assumed to be static, it is necessary for the robot to properly deal with the dynamic aspects of the environment. For example, a self driving car trying to cross an intersection in heavy traffic needs to be able to detect the individual dynamic objects in its vicinity such as cars, bikes, trucks and pedestrians. Furthermore it is necessary to estimate their individual motion characteristics to be able to navigate in a safe and efficient way. Understanding the dynamic nature of the environment offers many advantages. First, removing dynamic objects from a map can help to more accurately estimate the pose of a robot. Second, predicting the location of dynamic objects facilitates motion and path planning. Third, estimating the dynamics of objects can also help to infer semantic information.

In this work we propose a model-free approach for detecting and tracking dynamic objects in urban environments. Traditionally, model-free approaches rely on detecting dynamic objects by analyzing the perceived change of the environment caused by motion. Instead of detecting changes, we segment distinct objects using motion cues. Change detection either requires a prior map or an online mapping technique. In contrast, our approach does not require a map. We begin to reason about objects at point level by matching corresponding points in consecutive scans. This information is used to detect different motions in the scene. We recover the local static structure and multiple dynamic objects only based on the detected motion. Since we estimate motion models, we can reason about the dynamics of an detected object to efficiently track it. Tracking objects is a challenging problem since the sensor motion and the motion of objects lead to frequent occlusions, making data association a hard problem. Our framework leverages the estimated motion models for associating objects in consecutive frames.

The main contribution of our work is a novel approach for detecting and tracking dynamic objects in 3D LiDAR scans. We sequentially use RANSAC [4] to estimate motion models and propose a Bayesian approach to segment multiple objects. Our method only relies on motion cues and does not require any prior information about the object. Fig. 1 shows multiple objects of different types, shapes, and sizes detected and tracked by our approach.

II. RELATED WORK

The problem of detecting and tracking multiple moving objects has been studied actively for decades [2], [3]. Proposed methods to solve this problem can be broadly subdivided into model-free [6], [9], [16] and model-based [8], [11], [10] approaches.

In model-based approaches, objects are detected on the basis of known model information. These approaches are preferred when the object to be detected is known and therefore can be modeled a priori. In [8], an approach for detection and tracking of cars is presented. For people, Spinello et al. proposed a learning based approach [11]. They subdivide a human structure into multiple layers based on height and then learn a classifier for each layer. In [10], Shackleton et
al. outline another method for detecting and tracking people. The main disadvantage of model-based approaches is that they do not generalize to objects of different categories. To overcome this disadvantage, we propose a model-free approach for detecting generic objects.

Model-free methods are mainly based on motion cues and enable detection and tracking of objects of arbitrary shape and size. Since these approaches require motion information, they are unable to detect objects which can potentially move but are static in the current observation. Model-free approaches are generally based on building a static map of the scene and using this map information for detecting dynamic objects. In [9], Pomerleau et al. make a visibility assumption that the scene behind the object is observed, if an object moves. To leverage over this information, they compare an incoming scan with a global map and detect dynamic objects. Since they only use depth as cue for change detection, there method might fail if the motion between scans is small.

Kaestner et al. [6] propose a generative Bayesian approach for detecting dynamic objects. For tracking they use an approach based on the Kalman filter. They show results for a static sensor but as mentioned in Pomerleau et al. [9] there is no straightforward extension of their approach to a moving sensor. Recently, Wang et al. [16] proposed a model-free approach for detection and tracking in 2D LiDAR data. Using a joint state representation, they estimate the state of the sensor, a local static map, and the state of the dynamic object. Every incoming scan is associated with a local map and with dynamic objects. For tracking, they use a constant velocity motion model. While we have similar objectives, a comparison to our method is infeasible since our approach works on 3D instead of 2D LiDAR data.

Azim et al. [1] represent the environment using an octree-based occupancy grid and determine inconsistencies between the map and incoming scans to detect dynamic objects. In contrast, we do not build a map, but similar to [16] we only store local static information. For tracking they use Global Nearest Neighbor for associating tracks between consecutive frames. Tipaldi et al. [13] outline an approach for detecting and estimating motion using CRF for 2D LiDAR data. Van De Ven et al. [15] extended their approach by integrating the CRF based method with scan matching using a graphical model.

Moosmann et al. [7] use a segmentation method based on local convexity for detecting object hypotheses. They combine ICP and a Kalman filter for tracking and a classification method for managing tracks. Unlike them, we do not use a shape prior for detection but only rely on motion information. We compare our approach with their method and show superior performance. To best of our knowledge, this is one of the initial contribution for model-free detection and tracking in 3D LiDAR data.

III. FRAMEWORK OVERVIEW

The goal of our approach is to segment and track dynamic objects in LiDAR scans obtained by a mobile robot. Our framework shown in Fig. 2 consists of modules for detecting motion, tracking the sensor and tracking dynamic objects. A LiDAR scan is defined as a set of points:

$$P = \{ p_k \mid p_k \in \mathbb{R}^3, k = 1, \ldots, K \}$$

At every time step $t$, the scans $P_t$, $P_{t-1}$, and the motion models $T_{t-1} \in SE(3)$ are provided to the framework. In our notation $T_t$ describes the motion from $P_{t-1}$ to $P_t$. The points in $P_{t-1}$ are classified as either static, dynamic, or unknown point, i.e.: $P_{t-1} = P_{t-1}^S \cup P_{t-1}^D \cup P_{t-1}^U$.

First, the sensor tracking module classifies a subset of points $P_{t-1}^S$ as static and estimates the sensor motion $T_{t-1}^S$ relative to these points. Second, the object tracking module classifies a subset of points $P_{t-1}^D \subset (P_{t-1}^S \setminus P_t^S)$ as dynamic, where $P_{t-1}^D = P_{t-1}^D \cup \ldots \cup P_{t-1}^D_n$ consists of multiple disjoint point sets assigned to different dynamic objects. The module also estimates motion models $T_{t-1}^D = (T_{t-1}^{D_1}, \ldots, T_{t-1}^{D_n})$ for these objects. Third, all points $P_{t-1} \setminus (P_{t-1}^S \cup P_{t-1}^D)$ not classified by the tracking modules are provided to the detection module, which either adds them to the static point set, creates a new dynamic object, or assigns them to the unknown set $P_{t-1}^U$. For the first scan all points are unknown, i.e., $P_1 = P_1^U$.

IV. MOTION-BASED DETECTION

We segment dynamic objects using only motion cues. The motion of points between two consecutive LiDAR scans is mainly caused by the motion of the sensor and the motion of dynamic objects. We assume that the motion of these objects is rigid. The following subsection describes how we estimate motion models for the sensor and the dynamic objects. Subsequently, we explain the proposed Bayesian approach which calculates the probability of a point to follow a given motion model. Finally, we introduce our data association between consecutive scans and outline the method for detecting multiple motions. To simplify the notation in this section we assume without loss of generality that no points are classified already, i.e., $P_{t-1} = P_{t-1}^U$. 

---

Fig. 2. Our framework consists of modules for motion detection, tracking the sensor, and tracking dynamic objects. Point sets are indicated with $P$, motions models with $T$. 

---
A. Motion Models

We use RANSAC to estimate motion models \( T_t \in SE(3) \) for the sensor and the dynamic objects. To find initial point correspondences between the two scans, we uniformly sample keypoints \( F_{t-1} \subset P_{t-1} \), match their SHOT descriptors [14] against all points in \( P_t \), and pick the matches with minimum descriptor distance. We discard correspondences with a distance greater than a threshold, which is determined by an assumed motion limit and the sensor frame rate. RANSAC estimates the motion model \( T_t \) consented by the majority of the remaining correspondences. We define the inlier point set \( I_{t-1} \subset F_{t-1} \) as all points in \( P_{t-1} \) that are part of an inlier correspondence.

So far, \( T_t \) is only related to the motion of points in \( I_{t-1} \), which is a sparse subset of all the points in \( P_{t-1} \). Therefore it is necessary to infer all non-inlier points which also follow \( T_t \). In the next subsection we propose an approach to tackle this problem.

B. Bayesian Approach

We propose a Bayesian approach to determine the probability of each point \( p_k \in P_{t-1} \) to follow a given motion model. By this means we expand the sparse subset \( I_{t-1} \) to all points in scan \( P_{t-1} \). We represent the consent of \( p_k \) with \( T_t \) as a Bernoulli distributed random variable \( h_k \), where \( h_k = 1 \) means that \( p_k \) follows the motion model \( T_t \). The objective of the proposed Bayesian approach is to calculate the probability \( p(h_k \mid P_t, \hat{p}_k) \), where \( \hat{p}_k \in \hat{P}_{t-1} \) is the point \( p_k \) transformed by \( T_t \). We apply Bayes’ rule and assume independence of \( h_k \) and \( \hat{p}_k \) to calculate this probability:

\[
p(h_k \mid P_t, \hat{p}_k) = \frac{p(P_t \mid h_k, \hat{p}_k)p(h_k, \hat{p}_k)}{p(P_t \mid \hat{p}_k)p(\hat{p}_k)} \tag{3}
\]

\[
p(h_k \mid P_t, \hat{p}_k) = \frac{p(P_t \mid h_k, \hat{p}_k)p(h_k)p(\hat{p}_k)}{p(P_t \mid \hat{p}_k)p(\hat{p}_k)} \tag{4}
\]

\[
p(h_k \mid P_t, \hat{p}_k) \propto p(P_t \mid h_k, \hat{p}_k)p(h_k) \tag{5}
\]

In the following subsections we describe how we model the likelihood \( p(P_t \mid h_k, \hat{p}_k) \) and the prior \( p(h_k) \) in (5). The proposed Bayesian approach relies on the fact that \( \hat{p}_k \) is well aligned with points in \( P_t \), if \( h_k = 1 \). Therefore the depth, i.e. the distance to the sensor, of \( \hat{p}_k \) and the neighboring points in \( P_t \) should be similar. This is also true for local geometry. 

Furthermore, we impose by regularization that \( p_k \) is likely to follow the same motion as points in its vicinity.

1) Depth: The depth of the point \( p \) is denoted \( z \). Since the likelihood of \( h_k \) actually only depends on the neighborhood \( N_k \subset P_t \) of \( \hat{p}_k \) and we assume independence between the points \( p_l \in N_k \) we can model the likelihood in (5) based on the depth alignment [5] as

\[
p(P_t \mid h_k, \hat{p}_k) = \prod_{p_l \in N_k} p(z_l \mid h_k, \hat{z}_k) \tag{6}
\]

Similar to [9], we define \( N_k \) as the conical frustum around \( \hat{p}_k \). It better captures the disparity caused by misalignment in comparison to a spherical neighborhood.

To model the likelihood \( p(z_l \mid h_k, \hat{z}_k) \), we use the beam-based sensor model from [12]. For \( h_k = 1 \) it is a linear combination of four distributions:

\[
p(z_k \mid h_k = 1, \hat{z}_k) = \begin{pmatrix} w_{hit} \\ w_{short} \\ w_{max} \\ w_{rand} \end{pmatrix}^T \begin{pmatrix} P_{hit} \\ P_{short} \\ P_{max} \\ P_{rand} \end{pmatrix} \tag{7}
\]

where \( w_{hit} + w_{short} + w_{max} + w_{rand} = 1 \) and \( P_{hit} \) models the probability of hitting the expected surface, \( P_{short} \) of hitting an unexpected obstacle in front, \( P_{max} \) of a maximum range measurement, and \( P_{rand} \) of an unexplainable measurement.

\[
p_{hit} = \begin{cases} \eta \cdot \mathcal{N}(z_k; \hat{z}_k, \sigma^2) & \text{if } 0 \leq z_k \leq z_{max} \\ 0 & \text{otherwise} \end{cases} \tag{8}
\]

\[
p_{short} = \begin{cases} \eta \cdot \lambda_{short} e^{-\lambda_{short} z_k} & \text{if } 0 \leq z_k \leq \hat{z}_k \\ 0 & \text{otherwise} \end{cases} \tag{9}
\]

\[
p_{max} = \begin{cases} 1 & \text{if } z = z_{max} \\ 0 & \text{otherwise} \end{cases} \tag{10}
\]

\[
p_{rand} = \begin{cases} \frac{1}{z_{max}} & \text{if } 0 \leq z_k \leq z_{max} \\ 0 & \text{otherwise} \end{cases} \tag{11}
\]

For \( h_k = 0 \) the likelihood (7) changes to a mixture of three distributions:

\[
p(z_k \mid h_k = 0, \hat{z}_k) = \begin{pmatrix} w_{short} \\ w_{max} \\ w_{rand} \end{pmatrix}^T \begin{pmatrix} P_{short} \\ P_{max} \\ P_{rand} \end{pmatrix} \tag{12}
\]

where the weights are adapted and \( p_{short} \) in (9) is cut off at the maximum range \( z_{max} \) instead of the expected measurement \( \hat{z}_k \).

2) Local Geometry: We measure consistency of local geometry by calculating the cosine similarity \( c_{kl} \) between the SHOT descriptors \( f_k \) and \( f_l \):

\[
c_{kl} = \frac{\langle f_k, f_l \rangle}{\|f_k\| \|f_l\|} \tag{13}
\]

To incorporate the consistency of local geometry into the likelihood, we change (6) by weighting the individual factors with \( c_{kl} \):

\[
p(P_t \mid h_k, \hat{p}_k) = \prod_{p_l \in N_k} c_{kl} p(z_l \mid h_k, \hat{z}_k) \tag{14}
\]

3) Regularization: Since \( p_k \) likely follows the same motion as points in its vicinity, we impose a prior \( p(h_k) \) in (5). We model this prior by utilizing the information provided by RANSAC, namely if points in the neighborhood of \( p_k \) are in the inlier point set \( I_{t-1} \subset F_{t-1} \) or not. The neighborhood is calculated by drawing a sphere around \( p_k \):

\[
N^F_k = \{ p_l \in F_{t-1} \mid \|p_k - p_l\| < 3 \sigma_w \} \tag{15}
\]

Points \( p_k \) that have many inlier and few outlier points in their vicinity should have a high prior to consent with \( T_t \) and
vice versa. We realize these characteristics by computing a weighted average:

\[
p_0(h_k) = \begin{cases} 1 & \text{if } p_k \in I_{t-1} \\ 0 & \text{otherwise} \end{cases} \quad (16)
\]

\[
p(h_k) = \frac{\sum_{p_i \in N_I^k} w_{kl} p_0(h_k)}{\sum_{p_i \in N_I^k} w_{kl}} \quad (17)
\]

Since we want closer points to have a higher influence on the prior, we use a Gaussian for the weighting:

\[
w_{kl} = \frac{1}{\sqrt{2\pi\sigma_w^2}} \exp\left(\frac{||p_k - p_l||^2}{2\sigma_w^2}\right) \quad (18)
\]

The proposed Bayesian approach is used to calculate the probability of points in \( P_{t-1} \) to follow the motion model \( T_t \). The next subsection describes how we associate \( T_t \) with points in the current scan \( P_t \).

C. Data Association

To classify points in the current scan \( P_t \), we need to associate them to the corresponding motion models \( T_t \). Due to sensor motion and noise, every scan consists of varying points which are sampled from surfaces of the real structure. Therefore, establishing point-to-point correspondences between \( P_{t-1} \) and \( P_t \) is not feasible.

To realize the data association, we first identify points \( p_i^t \in P_{t-1} \subset P_{t-1} \) with a probability \( p(h_k = 1 | p_i, \hat{p}_k) > \zeta \), i.e., points that have a high probability to follow the motion model \( T_t \). Second, we determine the union of neighborhoods \( N \) by drawing spheres around all points \( \hat{p}_k^t \in \hat{P}_{t-1}^t \). Third, we declare points \( p_l \in P_l \) to follow the motion \( T_l \) if they lie inside \( N \). In this way we classify points in \( P_t \) according to their motion and assign them to the static point set \( P_s^t \) or to a point set of a dynamic object \( P_d^t \). We apply Euclidean clustering to handle multiple objects following the same motion.

D. Multiple Motions

We use RANSAC to detect motion in the scene (IV-A). To identify all points in scan \( P_t \) that follow a motion model \( T_t \), we use the Bayesian approach (IV-B) and apply the data association (IV-C). To detect multiple motions, we apply this pipeline sequentially. The detected \( T_t \) always stems from the motion which is consented by the majority of points. Therefore we remove all these points from \( P_{t-1} \) and \( P_t \) to subsequently determine the next motion model.

Since we assume that most points originate from static structure, the first \( T_t \) we detect is the sensor motion \( T_s^S \). We create a new dynamic object if

\[
||T_t^{D_1} - T_t^S||_F > \varepsilon \quad (19)
\]

where \( T_t^{D_1} \) is its estimated motion model, \( ||\cdot||_F \) the Frobenius norm, and \( \varepsilon \) a threshold that determines the minimum motion of an object to be considered as dynamic. If (19) is false, we add points to \( P_s^t \).

V. TRACKING THE SENSOR AND DYNAMIC OBJECTS

Tracking is concerned with propagating information over time, in our case from LiDAR scan \( P_{t-1} \) to \( P_t \). This involves updating the motion models of the sensor and the dynamic objects from \( T_{t-1} \) to \( T_t \). Furthermore, the points in \( P_t \) have to be classified as static or assigned to a specific dynamic object. In the following subsection we describe how we track the motion of the sensor and of dynamic objects.

A. Sensor Motion

We first update the motion model of the sensor from \( T_{t-1}^S \) to \( T_t^S \). To get an initial estimate how static points have moved, we apply the previous motion model:

\[
\tilde{T}_{t-1}^S = T_{t-1}^S * P_{t-1}^S \quad (20)
\]

Based on this estimate, we find correspondences between all points \( \tilde{p}_k \in \tilde{P}_{t-1}^t \) and points in \( P_t \) using nearest neighbor in Euclidean space. We apply RANSAC to estimate the motion \( T_t^S \) and update the motion model of the sensor:

\[
T_t^S = \tilde{T}_t^S * T_{t-1}^S \quad (21)
\]

To classify points in \( P_t \) as static, we first use the proposed Bayesian approach (IV-B). Based on the probabilities \( p(h_k \mid P_t, \hat{p}_k) \), we identify points that were static in \( P_{t-1} \) but are dynamic or disappeared in \( P_t \), i.e., have a low probability. Points with a high probability to consent with the sensor motion \( T_t^S \) are used in the data association (IV-C) to assign points to \( P_s^t \subset P_t \).

B. Dynamic Objects

The tracking of dynamic objects relies on a similar concepts as the sensor tracking. However, updating the motion model of an object from \( T_{t-1}^{D_1} \) to \( T_t^{D_1} \) is realized differently. First, we again apply the previous motion model to get an initial estimate where the object has moved:

\[
\tilde{T}_{t-1}^{D_1} = T_{t-1}^{D_1} * P_{t-1}^{D_1} \quad (22)
\]

We determine a neighborhood \( N \subset P_t \) as it is done in (IV-C). This provides a coarse prior information, where to search for correspondences, namely between points in \( P_{t-1}^{D_1} \) and \( N \). We find correspondences by matching SHOT descriptors, which are subsequently used by RANSAC to estimate the motion model \( T_t^{D_1} \). To assign object points to \( P_{t-1}^{D_1} \subset P_t \), we choose the same approach as for the sensor tracking.

We create a tracklet for every segmented dynamic object, which is defined by its motion model \( T_t^{D_1} \) and point set \( P_{t-1}^{D_1} \). Tracklets tracked for more than \( N_T \) scans are promoted as tracks. We make this distinction to avoid false positives. A track is lost when an object is no longer in the sensors field of view or when it is entirely occluded, i.e., \( P_{t-1}^{D_1} = \emptyset \). To tackle temporary occlusions we predict a bounding box based on the objects last observation. We recover a track when points reappear inside the tracks bounding box.
VI. RESULTS

We evaluate our approach on two datasets made available by [7]. Both scan sequences A and B are collected in non-flat urban environments using a Velodyne HDL-64E LiDAR sensor at 10Hz and are 38 and 50 seconds long, respectively. The ground truth velocity of the sensor was obtained using DGPS/IMU and is also available for one other car. To conduct an extensive quantitative analysis, we manually labeled all dynamic objects apart from pedestrians and compared the results of our approach against the Moving Object Mapping (MOM) method presented by Moosmann et al. [7]. For all experiments presented in this section, we chose $\zeta = 0.95$, $\varepsilon = 0.2$, and $N_T = 8$.

Fig. 3 shows a snapshot of sequence B taken at $t = 26.2$. Since we remove the ground plane before we provide a scan to the framework, ground points are not classified. It can be seen that we are able to segment and track many objects of various types, e.g. cars, trucks, and bikes. This is one major advantage of our approach compared to model-based methods. Admittedly, our approach does not work well for pedestrians. This is mainly for the reason that they move slowly and our detection method only relies on motion cues. Furthermore, the assumption of rigid body motion does not hold. Due to their comparatively small size, pedestrians may also consist of very few points, especially if they are far away from the sensor. Since pedestrians are not labeled in the ground truth, we ensure not to count them as false positives in the analysis.

Table I reports results for our approach and compares them. In sequence A we clearly outperform MOM which suffers from a high number of false positives. Its different recall and object recall values are caused by the dominance of one object, that in contrast to others, is present over the whole sequence and can be tracked. In sequence B our approach reaches significantly better precision with a slightly worse recall. In both sequences we achieve better $F_1$ scores.

For a quantitative evaluation of our approach, we use the ground truth bounding boxes of the dynamic objects. For each scan we compute precision and recall, where precision is defined as the ratio between the number of correct detections and detections and recall as the ratio between the number of correct detections and bounding boxes. We average precision and recall over all scans of the sequence and compute the $F_1$ score. We also define an object recall (ObjRcl) that measures the ratio between the number of scans in which an object is detected and scans in which it is actually there according to the ground truth. We average this value over all objects. In contrast to recall, every object equally contributes to the object recall, independent on the number of scans in which the object is present.

Table I reports results for our approach and compares them. In sequence A we clearly outperform MOM which suffers from a high number of false positives. Its different recall and object recall values are caused by the dominance of one object, that in contrast to others, is present over the whole sequence and can be tracked. In sequence B our approach reaches significantly better precision with a slightly worse recall. In both sequences we achieve better $F_1$ scores.

Fig. 3 shows a snapshot of sequence B taken at $t = 26.2$. Since we remove the ground plane before we provide a scan to the framework, ground points are not classified. It can be seen that we are able to segment and track many objects of various types, e.g. cars, trucks, and bikes. This is one major advantage of our approach compared to model-based methods. Admittedly, our approach does not work well for pedestrians. This is mainly for the reason that they move slowly and our detection method only relies on motion cues. Furthermore, the assumption of rigid body motion does not hold. Due to their comparatively small size, pedestrians may also consist of very few points, especially if they are far away from the sensor. Since pedestrians are not labeled in the ground truth, we ensure not to count them as false positives in the analysis.

<table>
<thead>
<tr>
<th>Classification of Dynamic Objects</th>
<th>Precision</th>
<th>Recall</th>
<th>$F_1$</th>
<th>ObjRcl</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sequence A</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td>62.41</td>
<td>91.33</td>
<td>70.76</td>
<td>81.77</td>
</tr>
<tr>
<td>MOM</td>
<td>14.89</td>
<td>72.91</td>
<td>22.02</td>
<td>30.90</td>
</tr>
<tr>
<td><strong>Sequence B</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td>72.57</td>
<td>78.18</td>
<td>73.05</td>
<td>74.08</td>
</tr>
<tr>
<td>MOM</td>
<td>29.85</td>
<td>89.01</td>
<td>41.81</td>
<td>79.91</td>
</tr>
</tbody>
</table>
absolute speed of another car. Therefore we evaluate the estimation of the sensor motion and the relative motion between sensor and dynamic object. Fig. 5 shows that we can robustly track both in sequence A. The peak in the sensor motion is caused by a corrupted LiDAR scan. In Table II we report the median, mean, standard deviation, and RMSE generated without the outer 10%-quantiles for the sensor motion in both sequences. The table also shows results for the dynamic objects tracking in comparison to the results reported in Moosmann et al. [7]. It can be seen that our approach outperforms MOM. To provide a more informative measure we also report the RMSE for our approach.

<table>
<thead>
<tr>
<th>Sensor Speed Error</th>
<th>Object Speed Error</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Median</strong> A: -0.19</td>
<td><strong>Median</strong> MOM: -0.84</td>
</tr>
<tr>
<td><strong>Mean</strong> A: -0.09</td>
<td><strong>Mean</strong> MOM: -0.69</td>
</tr>
<tr>
<td>Std.-Dev. A: 0.57</td>
<td>Std.-Dev. MOM: ±1.16</td>
</tr>
<tr>
<td>RMSE</td>
<td>RMSE</td>
</tr>
<tr>
<td>0.30</td>
<td>0.75</td>
</tr>
</tbody>
</table>

**TABLE II**

**ESTIMATION OF SENSOR AND DYNAMIC OBJECT MOTION**

**GENERATED WITHOUT OUT 10%-QUANTILES**

VII. CONCLUSIONS

In this paper we present a novel approach to detect and track dynamic objects. We detect motions between consecutive scans by sequentially using RANSAC and propose a Bayesian approach to segment and track multiple objects. Our method is model-free, i.e., it does not require any prior information about the objects. We analyze our approach on two publicly available data sequences and compare it with an existing method. For both sequences, our approach achieves a better $F_1$ score. Furthermore, we show that we track the speed of the sensor and of another object with a higher accuracy.

ACKNOWLEDGEMENT

We would like to thank Pratik Agarwal for the fruitful discussions and Frank Moosmann for providing the dataset and the source code for Moving Object Mapping [7].

REFERENCES


