Moving Vehicle Detection and Tracking in Unstructured Environments

Nicolai Wojke, Marcel Häselich
Active Vision Group, AGAS Robotics
University of Koblenz-Landau
Universitätsstr. 1
56070 Koblenz, Germany
{nwojke,mhaeselich}@uni-koblenz.de

***Abstract*— The detection and tracking of moving vehicles is a necessity for collision-free navigation. In natural unstructured environments, motion-based detection is challenging due to low signal to noise ratio. This paper describes our approach for a 14 km/h fast autonomous outdoor robot that is equipped with a *Velodyne HDL-64E S2* for environment perception. We extend existing work that has proven reliable in urban environments. To overcome the unavailability of road network information for background separation, we introduce a foreground model that incorporates geometric as well as temporal cues. Local shape estimates successfully guide vehicle localization. Extensive evaluation shows that the system works reliably and efficiently in various outdoor scenarios without any prior knowledge about the road network. Experiments with our own sensor as well as on publicly available data from the DARPA Urban Challenge revealed more than 96 % correctly identified vehicles.**

I. INTRODUCTION

The detection and tracking of dynamic objects is an essential task for collision-free autonomous navigation of outdoor robots. Compared to urban areas, where road network information is available for background separation and motion prediction, unstructured environments present a more challenging scenario due to low signal to noise ratio.

We present an approach to vehicle detection and tracking that is independent of any prior knowledge of the environment, in particular of the road network. The approach handles vehicles of various appearance, size, and speed as well as partial and full occlusions over short periods of time. For environment perception, we rely on modern 3D laser range finders (LRFs). They provide rich 3D point clouds with full 360° perception of large distances around the robot. For this work we use a *Velodyne HDL-64E S2* with an update frequency of up to 15 Hz. Therefore, updates arrive every 66 ms, allowing algorithms to detect changes in split second.

We extend an existing approach of Petrovskaya and Thrun [19] that was designed for the DARPA Urban Challenge in 2007. The approach heavily relies on road network information for background subtraction to reduce the number of false positive detections and to reduce the computational load of continuing vehicle hypothesis verification. We extend their work to unstructured environments without limiting search space to areas of likely vehicle occurrence. Therefore, we run foreground separation using temporal and geometric cues. In order to deal with sensor noise and sporadic range readings we extend the two dimensional virtual scan of Petrovskaya and Thrun [19] to a Gaussian environment model that captures the local point distribution in the neighborhood of each reading. Following prior work [22], [26] we use line and corner features to guide vehicle localization in areas of change. Vehicle tracking is carried out using a Rao-Blackwellized Particle Filter as in [19].

This paper is organized as follows. Sec. II starts with a discussion of previous related work. Sec. III encapsulates the work we extend in our approach. Sec. IV and Sec. V describe our implementation. Finally, Sec. VI depicts our experimental results and Sec. VII presents our conclusion.

II. RELATED WORK

Over the past few decades, a large number of detection and tracking approaches have been developed, especially in the context of the DARPA Grand and Urban Challenges. Existing approaches can be divided into appearance-based and motion-based methods.

Appearance-based methods on the one hand allow for an identification of dynamic objects even if they are currently not moving. Sun et al. [20] give a survey on vision-based vehicle detection and tracking over the last 15 years. Most vision-based approaches locate possible vehicles based on visual cues. Hypothesis are then validated using template-matching or binary classification.

During the Grand and Urban Challenges, Broggi et al. [3] used a wide baseline trinocular camera system to generate range images of up to $50\,\mathrm{m}$ in front of their robot. After filtering ground readings, remaining points were clustered to generate object hypothesis. During the VisLab Intercontinental Challenge the robot “Porter” was equipped with two stereo camera systems [2]. The cameras provided information for line marking, vehicle and pedestrian detection. Both systems were supported by multiple LRFs for terrain analysis and long range obstacle detection.

Wender and Dietmayer [25] generate vehicle hypothesis in planar laser scans by fitting corner features to point clusters. For verification in camera images, two separate classifiers for lateral views and for frontal views of vehicles are trained. The transformation is obtained from the candidate’s corner approximation in the planar laser scan.

In the approach described by Miller et al. [15] the output of a commercial vision-based vehicle tracking system is integrated into a multi-sensor environment representation.

Camera images are used for candidate generation rather than for validation.

Dietmayer et al. [7] use only LRF data to detect traffic participants exclusively based on their geometry. They obtain geometric extents by fitting oriented bounding boxes to point clusters in planar laser scans.

Morris et al. [17] use a combination of 2D and 3D scans to identify vehicles in cluttered environments. The 2D scans are used for hypothesis generation which are further examined in the 3D scans. A linear support vector machine is trained to discriminate vehicles from clutter.

Motion-based methods on the other hand create candidates in areas of change and therefore only detect moving objects. Explicit discrimination based on appearance is avoided to overcome high intra-class variance. Instead, motion is used as predominant feature for object detection.

Wang [23] detects moving objects as a pre-filter for simultaneous localization and mapping. The environment is represented by two distinct occupancy grid maps: one for static and one for dynamic objects. In a related project Wang et al. [24] use a motion-based approach to detect pedestrians and vehicles in planar laser scans. For object tracking an extended Kalman filter with Multiple Hypothesis Tracking [6] is used.

Coué et al. [5] provide a Bayesian formulation of occupancy maps that associates a velocity with each grid cell. Cell movements are predicted using a linear velocity model. Gindele et al. [10] formulate *Bayesian Occupancy Filter Using Prior Map Knowledge* (BOFUM) to incorporate a priori knowledge about the street layout. In order to reduce computational expenses for BOFUM, Brechtel et al. [1] introduce a probabilistic sampling based solver.

A comprehensive collection of the scientific results obtained at the DARPA Urban Challenge by participating finalist teams is given in [4]. The robot “Boss” of team Tartan Racing [21] is equipped with a combination of in total 18 laser and radar sensors, including a Velodyne HDL-64. Lasers are used to cluster points, whereas the radars are used for speed estimation. In close and medium range line and corner features are fit to object clusters. Depending on the geometric approximation objects are tracked using the bicycle model [13] or a constant acceleration model.

Team MIT [14] uses a combination of 12 planar LRFs and a Velodyne HDL-64E for environment perception. Additionally 15 radar sensors cover the area around the vehicle to obtain precise velocity estimates in long distances. The environment is represented by an array of cells where each cell stores information about the occupancy of the area it represents.

The Stanford Racing Team [16] is another participant of the DARPA Urban Challenge. Petrovskaya and Thrun [19] use a model-based approach to vehicle detection and tracking from 3D laser data alone by extracting a two dimensional planar scan. Their model is used for precise motion estimation instead of vehicle classification and their approach does not require data association.

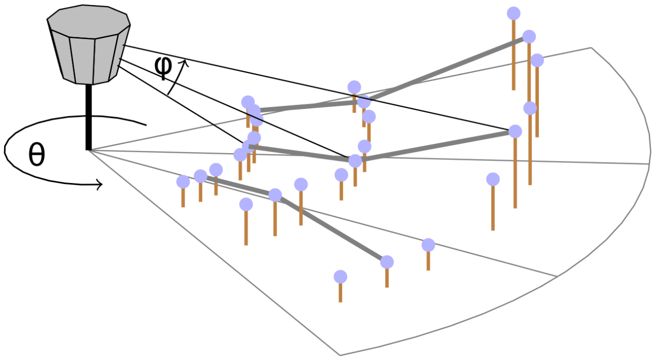

Fig. 1: Projection of Velodyne HDL-64E readings into a 3D grid in spherical coordinates as described in [19]. Readings are illustrated as dots. Their height is indicated by a line that projects readings into the horizontal plane.

III. SENSOR DATA INTERPRETATION

Our system is based on the approach of Petrovskaya and Thrun [19]. The authors introduce a measurement model for vehicle tracking with planar LRF that allows direct interpretation of range data. This section describes the measurement model as well as the method to generate a two dimensional scan representation from data of a Velodyne HDL-64E.

A. *Sensor Data Representation*

In nearly planar environments, vehicle tracking is a problem in two dimensional space. However, the 3D data of the Velodyne HDL-64E can be used to filter ground readings as perceived by planar LRFs in uneven terrain. Petrovskaya and Thrun [19] describe the extraction of a two dimensional virtual scan from data of a Velodyne HDL-64E with a simple ground model. Therefore, all readings are projected into a 3D grid in polar coordinates around the vehicle center (cf. Fig. 1). For filtering noise, a representative reading is chosen for each cell as the median of distances. A model of ground elevation is established by comparing neighboring cells with the same horizontal bearing θ. If the slope is smaller than a threshold, the point is classified as ground reading. The distance to the closest obstacle in a target height from 50 cm to 200 cm is used as entry in the virtual scan.

B. *Measurement Model*

The measurement model [19] approximates vehicle geometry as rectangular shape of non-zero depth (cf. Fig. 2). The likelihood of range readings is modeled as piecewise constant function over the domain of range readings $[r_{\min}, r_{\max})$. Naturally, the likelihood of a reading to fall onto the vehicle surface is highest. Due to self-occlusion, there are at maximum two sides visible at a time. Rays that extent through the visible car surface into and beyond the vehicle interior are assigned a low likelihood. Similarly, range readings that fall short, but not in the close vicinity of the vehicle, receive a penalty. However, the likelihood of short readings remains high, as occlusion is common in dynamic environments, caused by objects such as street

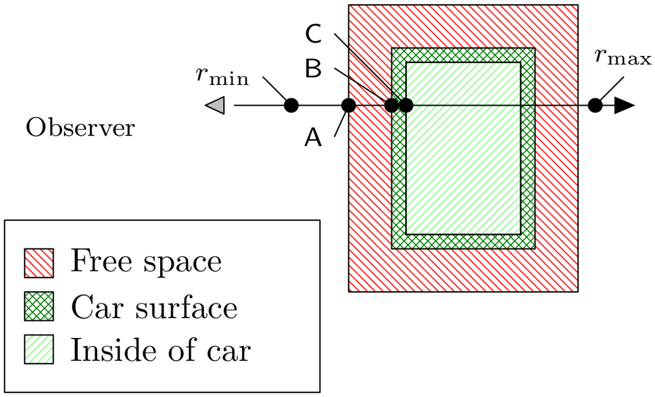

Fig. 2: Geometric regions involved in measurement model computation as described by Petrovskaya and Thrun [19].

signs, pedestrians, or other vehicles. By assigning a minimum likelihood to readings in the close vicinity of the vehicle, the measurement model enforces a region of free space.

IV. FOREGROUND SEPARATION

Natural unstructured environments are characterized by low signal to noise ratio. While LRFs provide precise geometric measurements in indoor environments, in the outdoors vegetation causes sporadic and large changes due to its scattered appearance. Additionally, environmental factors such as wind cause small and continuous changes. Errors in data reduction of full 3D point clouds to 2D virtual scans reinforce the effect.

While Petrovskaya and Thrun [19] restrict vehicle detection to the area close to the road, we have no road network information available to reduce the number of false positive detections and to reduce the computational load of continuing vehicle hypothesis verification. Consequently, we introduce a foreground model that incorporates geometric as well as temporal cues. With interest in moving vehicles, we define the foreground as all points that fall onto approximate piecewise planar object structures that exhibit motion.

A. Temporal Cues

Movement is perceived as change in the environment. To distinguish movement from background noise, we create a model of Gaussian distributions. Let z_i be the range reading of the i-th ray in the virtual scan. Then z_i is the median of set of readings $C_{i,j}$ of the j-th cell in the i-th cone of the spherical grid, where j is the index of the cell with the closest obstacle. We compute the normal point distribution from the set of readings $C_{i,j}$ and its eight-neighborhood in the spherical grid to obtain samples $X_t \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$ at time t for all rays in the virtual scan.

At time steps $t-1$ and t, we compare samples X_{t-1} and X_t to detect change. If sample X_{t-1} is classified as background and sample X_t is classified as foreground, we accept a change in the environment. If both samples are classified as foreground, we compare their means using the squared Mahalanobis distance

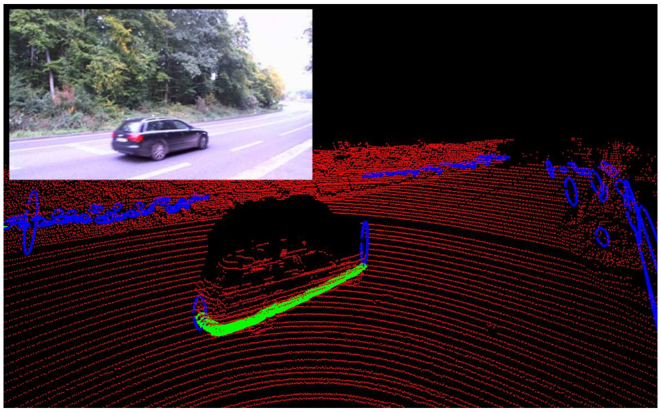

Fig. 3: A vehicle on a rural highway as perceived in the model of Gaussian distributions. Covariances of local point distributions are drawn as ellipses (blue: background, green: foreground).

$$d^2 = (\boldsymbol{\mu_t} - \boldsymbol{\mu_{t-1}})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu_t} - \boldsymbol{\mu_{t-1}})$$

with pooled covariance matrix

$$\boldsymbol{\Sigma} = \frac{N_t}{N_t + N_{t-1}} \boldsymbol{\Sigma_t} + \frac{N_{t-1}}{N_t + N_{t-1}} \boldsymbol{\Sigma_{t-1}},$$

where N_t is the size of sample X_t. A change in the environment is accepted if d^2 is greater than a threshold t_d.

To account for situations where the normal point distribution is an imprecise approximation, we threshold the length of the largest eigenvalue of the covariance matrix. A sample is classified as background if the largest eigenvalue of covariance $\boldsymbol{\Sigma}_t$ is larger than a threshold t_{σ^2}. Therefore, translucent and scattered objects are not considered in vehicle detection.

Fig. 3 shows a vehicle on a rural highway. The vehicle has a piece-wise planar surface with high point density and is therefore considered as foreground. Vegetation and clutter in the distance are classified as background due to their scattered appearance. Note that the Gaussian environment model is computed from the data of the 3D spherical grid rather than the 2D virtual scan. Accordingly, planar structures in the 2D virtual scan are not considered foreground if readings sporadically hit the surface (e.g., a mid-height fence).

B. Geometric Cues

We use line and corner features to describe local geometry. They support the underlying rectangular geometry of the measurement model. To acquire local shape estimates, we run region growing segmentation in areas of change. We then fit line and corner features using a RANSAC-based [8] approach. The following two models are fit to data:

Two-sided model: Due to self-occlusion there are at maximum two sides visible at a time. We use a set of two perpendicular lines to describe corner configurations.

One-sided model: If there is only one side visible, we use a single line to describe all readings within the point cluster.

If the data does not fit either of the two models, the point cluster is disregarded.

Fitting a line $\boldsymbol{l} = (n_x, n_y, -d)^{\mathrm{T}}$ with normal $\boldsymbol{n} = (n_x, n_y)^{\mathrm{T}}$ and distance to origin d requires two points. As error function the absolute Euclidean distance

$$\rho(\overline{\boldsymbol{p}}, \boldsymbol{l}) = |\overline{\boldsymbol{p}} \cdot \boldsymbol{l}| \tag{1}$$

between point $\overline{\boldsymbol{p}} = (p_x, p_y, 1)^{\mathrm{T}}$ and line $\boldsymbol{l}$ is used. Corner estimation requires three non-collinear points. The first two points define a line $\boldsymbol{l_1}$. The third point defines the position of the orthogonal line $\boldsymbol{l_2}$ that passes through the point and its perpendicular foot on $\boldsymbol{l_1}$. The error is computed as the minimum error between $\overline{\boldsymbol{p}}$ and lines $(\boldsymbol{l_1}, \boldsymbol{l_2})$:

$$\rho(\overline{\boldsymbol{p}}, (\boldsymbol{l_1}, \boldsymbol{l_2})) = \min\{\, \rho(\overline{\boldsymbol{p}}, \boldsymbol{l_1}), \rho(\overline{\boldsymbol{p}}, \boldsymbol{l_2}) \,\}. \tag{2}$$

The error threshold is configured to equal the vehicle surface width of the measurement model. RANSAC is configured to terminate according to the adaptive stopping criterion where the number of iterations is updated based on the highest inlier ratio observed so far [11].

Note that the one-sided model is a degenerate case of the two-sided model, where the position of the shorter side is free. We use the framework of RANSAC for (Quasi-)Degenerate Data (QDEGSAC) [9] to deal with quasi-degenerate configurations where few readings fall onto the shorter side. In a series of RANSAC runs QDEGSAC identifies degenerate configurations and selects the best fitting model based on the ratio of inliers to outliers.

V. Vehicle Detection and Tracking

The success of previous studies encourages the use of motion as predominant feature for the detection of moving objects. Therefore, we use a motion-based approach for vehicle detection and tracking in unstructured environments.

First, we separate vehicle candidates from background as explained in Sec. IV. Then, we localize candidates that are not yet explained by an existing track. In two consecutive frames, candidates are validated based on their motion pattern. Tracking is carried out using a Rao-Blackwellized Particle Filter as described in [19].

A. Vehicle Localization with Shape Priors

Given local shape estimates from the foreground test, vehicle localization is necessary to handle partial occlusions where the object is split up into multiple clusters. Similar to [19], we use Scaling Series [18] for vehicle localization. In order to derive a more relaxed version of the measurement model, the vehicle surface width is inflated proportional to the sample radius.

With local shape estimates available, the vehicle position and orientation (pose) is roughly known. For corner configurations, the pose is known up to a $90°$ ambiguity in orientation. For line configurations, we additionally search for the position along the given shape estimate.

We define the search space with two δ-spheres that resolve the $90°$ ambiguity in orientation. The vehicle surface width is initially padded 56.25 cm to each side. This is half the width used by Petrovskaya and Thrun [19]. With an approximation of shape available, it is not necessary to consume the entire free space in early iterations, because position and orientation are roughly known. In total, the algorithm runs 4 iterations until the target vehicle surface width of 25.0 cm is reached.

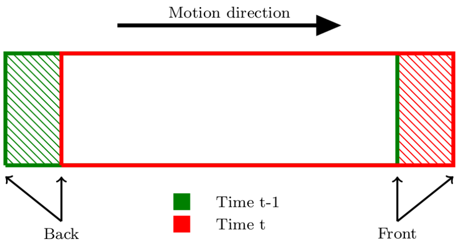

Fig. 4: Areas involved in *motion evidence* computation [19].

B. Validation

For track validation, we use the *motion evidence* score of Petrovskaya and Thrun [19]. The motion evidence score describes how likely change was caused by a moving vehicle, considering the areas that altered due to movement. Fig. 4 shows a moving vehicle in time steps $t-1$ and t. The area in front of the vehicle at time $t-1$ was previously free and became occupied. The area behind the vehicle in time step t was previously occupied and became free.

Note that the motion evidence score requires knowledge of the candidate's speed. However, when a candidate is first localized, the motion pattern can be tested against a minimum velocity v_{min} to quickly rule out changes that do not fit the model.

C. Tracking

For vehicle tracking, we use a Rao-Blackwellized Particle Filter that samples directly from the measurement model. For a detailed description we refer the reader to the original paper [19]. We found that spurious measurements in the background are not hindering tracking performance. Therefore, we use all points in the virtual scan for vehicle tracking, including points that were previously classified as background.

VI. Experiments

The system was evaluated with regard to detection times in ranges 0 m–30 m, 30 m–50 m, and 50 m–80 m. The test sequence contains in total eight vehicles that drive towards the ego-vehicle with velocities between 60–80 km/h. To enable multiple detections in different ranges, the tracker was configured to drop tracks after one frame. Table I summarizes the results. In the range of 50 m–80 m only 2 out of 8 vehicles were detected. Detection times ranged between 10 and 12 frames. Due to the limited angular resolution of the *Velodyne HDL-64E S2*, only few readings fall onto the vehicle surface in long distances. Further, inaccuracies in ground detection cause noisy measurements that are disregarded by the foreground model. In ranges 0 m–30 m and 30 m–50 m all vehicles were successfully detected. Detection times averaged between 3 and 4 frames. The maximum detection

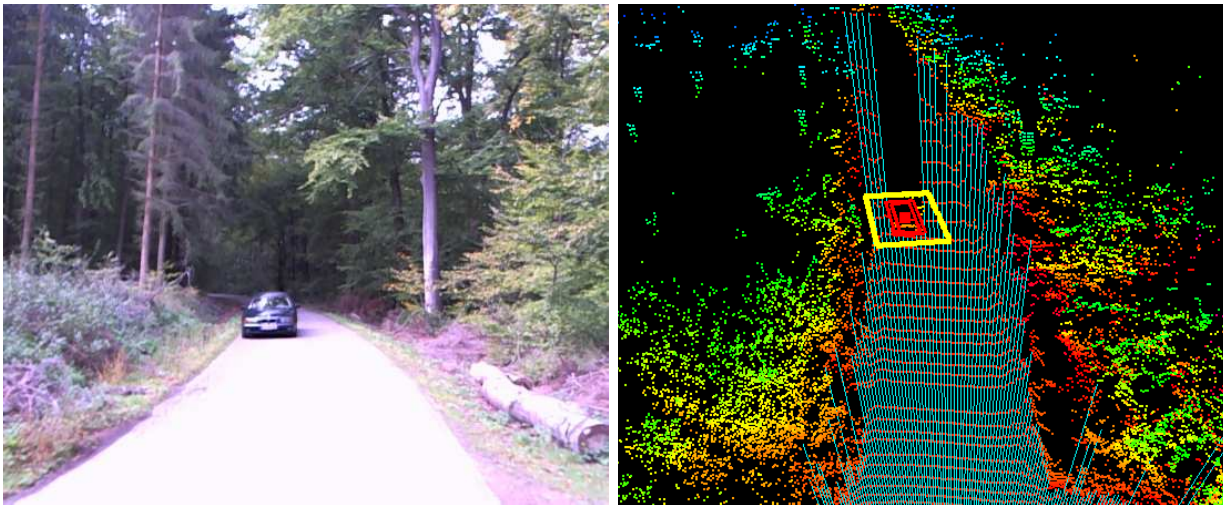

Fig. 5: Camera view and laser data with virtual scan of the detection and tracking of a dynamic obstacle. The example scenario shows the successful detection and tracking of a driving vehicle (yellow) in a difficult unstructured forest region. The 3D laser points are colored according to their height value (red - zero, yellow - 1 m, green - 2 m, and blue - over 2 m).

	80 m–50 m	50 m–30 m	30 m–0 m
Average detection time	11.33	3.50	3.38
Minimum detection time	10	3	3
Maximum detection time	12	5	5
Number of misses	6	0	0

TABLE I: Number of frames required for vehicle detection and number of total misses in 0 m–30 m, 30 m–50 m, and 50 m–80 m.

Mission	Duration	Distance	Detected	Missed
2	5,428 s	21.5 km	153	6
3	10,414 s	41.2 km	218	4

TABLE II: Summary of overall performance on the DARPA Urban Challenge data set of Huang et al. [12].

	Run-time: mean / standard deviation in ms				
Scenario	Scan	Detection	Validation	Tracking	Total
City center	23 / 3	5 / 3	12 / 13	13 / 9	54 / 15
Forest	26 / 4	12 / 5	15 / 14	0 / 0	53 / 16

TABLE III: Mean and standard deviation of run-times in a city and in a forest scenario.

time was 5 frames for both ranges. With at least 3 frames required for vehicle detection the system performed close to the theoretical minimum in ranges up to 50 m.

We further evaluated the overall performance on the publicly available DARPA Urban Challenge data set [12]. The data set pictures the DARPA Urban Challenge finals from the perspective of team MIT's robot "Talos". Evaluation was carried out by counting total number of detections and misses on mission files 2 and 3. Both missions sum up to in total 4 hours and 40 minutes autonomous driving over a distance of 62.7 km. Vehicles were considered for detection if they were within 50 m of the robot and visible in the virtual scan, i.e., not fully occluded by other objects. Moving vehicles were counted as missed in two situations: 1. if the vehicle was not detected and 2. if the track was not maintained during close contact. Specifically, vehicles must be tracked when entering an intersection and when passing the ego-vehicle in close range. Qualitative measures, such as minimum and maximum detection times, were omitted due to missing ground truth.

Table II summarizes the results. In total 361 out of 371 vehicles were correctly identified and tracked. Several misses occurred where vehicles were driving in the opposite direction on a divided highway. The view onto the other side of the road was obstructed by evenly spaced trees on the median. Consequently, vehicles were partially occluded with only few readings falling onto the vehicle surface. Further, we found that the initial detection of slow moving objects is challenging. Several misses were caused by slow moving traffic in the middle of the road or close to an intersection. Near misses occurred at intersections with overly careful approaching vehicles. Generally, these vehicles were detected on entering rather than on approaching the intersection. However, once these vehicles were detected tracking remained stable. Occasionally, the robot approached an intersection where another vehicle was waiting already. As vehicle localization is initiated in areas of change, the vehicle was not detected up until moving off the stop line.

For run-time evaluation we analyzed two data sets. The first data set was recorded during a trip through the congested city center of Koblenz, Germany. It is characterized by frequent traffic and partial as well as full occlusions, with up to ten vehicles being tracked simultaneously. The second data set was recorded in a natural forest environment. The data is characterized by vegetation and clutter. However, there was no encounter with other traffic participants. Table III summarizes run-times on a 2.53 GHz Intel Core 2 Duo. The system performed equally well in both scenarios with run-times of in average under 55 ms/frame, which is lower than the update frequency of the *Velodyne HDL-64E S2*.

VII. CONCLUSION AND FUTURE WORK

In this paper we successfully extended the approach of Petrovskaya and Thrun [19] towards unstructured environments. We managed to detect and track vehicles exclusively on the 3D data of a Velodyne HDL-64E S2 in real-time. The approach works without any prior knowledge about the environment and is able to deal with ego-motion even in rough terrain. In order to model changes caused by vegetation and clutter, change detection incorporated the distribution of points in the local neighborhood of each reading. Geometry was approximated using line and corner features. Evaluation on our own as well as publicly available data showed that the algorithms work reliably. Detection of fast moving vehicles performed close to the theoretical minimum of 3 frames within a range of 50 m. Overall more than $96\,\%$ of all vehicles were correctly identified. The tracker handled sharp corners, hard accelerations, and partial as well as full occlusions of up to two seconds. The presented approach is currently limited to vehicles. A possible extension could include the detection and tracking of other dynamic objects, such as pedestrians. In close range it is often beneficial to know about stationary objects as well to reason about their future behavior. A temporarily parked car may start moving at any time. Therefore, the 3D data could be used for appearance-based object detection in close range. Integration of further sensors represents another possible extension.

VIII. ACKNOWLEDGMENTS

This work was partially funded by Wehrtechnische Dienststelle 51 (WTD), Koblenz, Germany.

REFERENCES

[1] S. Brechtel, T. Gindele, and R. Dillmann. Recursive importance sampling for efficient grid-based occupancy filtering in dynamic environments. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3932–3938, 2010.

[2] A. Broggi, L. Bombini, C. Stefano, P. Cerri, and R. I. Fedriga. Sensing requirements for a 13,000 km intercontinental autonomous drive. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 500–505, 2010.

[3] A. Broggi, A. Cappalunga, C. Caraffi, S. Cattani, S. Ghidoni, P. Grisleri, P. P. Porta, M. Posterli, and P. Zani. TerraMax vision at the Urban Challenge 2007. *IEEE Transactions on Intelligent Transportation Systems*, 11(1):194–205, 2010.

[4] M. Buehler, K. Iagnemma, and S. Singh, editors. *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*, volume 56 of *Springer Tracts in Advanced Robotics*. Springer Press, 2009.

[5] C. Coué, C. Pradalier, C. Laugier, T. Fraichard, and P. Bessière. Bayesian occupancy filtering for multitarget tracking: An automotive application. *International Journal of Robotic Research*, 25(1):19–30, 2006.

[6] I. Cox and S. Hingorani. An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(2):138–150, 1996.

[7] K. Dietmayer, J. Sparbert, and D. Streller. Model based object classification and object tracking in traffic scenes from range images. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 25–30, 2001.

[8] M. Fischler and R. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[9] J.-M. Frahm and M. Pollefeys. Ransac for (quasi-)degenerate data (QDEGSAC). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 453–460, 2006.

[10] T. Gindele, S. Brechtel, J. Schröder, and R. Dillmann. Bayesian occupancy grid filter for dynamic environments using prior map knowledge. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 669–676, 2009.

[11] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision (2nd Edition)*. Cambridge University Press, 2003.

[12] A. Huang, M. Antone, E. Olson, L. Fletcher, D. Moore, S. Teller, and J. Leonard. A high-rate, heterogeneous data set from the DARPA Urban Challenge. *International Journal of Robotics Research*, 29(13):1595–1601, 2011.

[13] N. Kämpchen, T. Weiss, M. Schaefer, and Dietmayer.K. IMM object tracking for high dynamic driving maneuvers. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 825–830, 2004.

[14] J. Leonard, J. How, S. Teller, M. Berger, S. Campbell, G. Fiore, L. Fletcher, E. Frazzoli, A. Huang, S. Karaman, O. Koch, Y. Kuwata, D. Moore, E. Olson, S. Peters, J. Teo, R. Truax, M. Walter, D. Barrett, A. Epstein, K. Maheloni, K. Moyer, T. Jones, R. Buckley, M. Antone, R. Galejs, S. Krishnamurthy, and J. Williams. A perception-driven autonomous urban vehicle. *Journal of Field Robotics*, 25(10):727–774, 2008.

[15] I. Miller, M. Campbell, D. Huttenlocher, A. Nathan, F. Kline, P. Moran, N. Zych, B. Schimpf, S. Lupashin, E. Garcia, J. Catlin, M. Kurdziel, and H. Fujishima. Team cornell's skynet: Robust perception and planning in an urban environment. In *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*, pages 257–304, 2009.

[16] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Hhnel, T. Hilden, G. Hoffmann, B. Huhnke, D. Johnston, S. Klumpp, D. Langer, A. Levandowski, J. Levinson, J. Marcil, D. Orenstein, J. Paefgen, I. Penny, A. Petrovskaya, M. Pflueger, G. Stanek, D. Stavens, A. Vogt, and Sebastian Thrun. Junior: The Stanford entry in the Urban Challenge. *Journal of Field Robotics*, 25(9):569–597, 2008.

[17] D. Morris, R. Hoffman, and S. McLean. Ladar-based vehicle detection and tracking in cluttered environments. In *Proceedings of the 26th Army Science Conference*, 2008.

[18] A. Petrovskaya, O. Khatib, S. Thrun, and A. Y. Ng. Touch based perception for object manipulation. In *Proceedings of Robotics: Science and Systems*, June 2007.

[19] A. Petrovskaya and S. Thrun. Model based vehicle detection and tracking for autonomous urban driving. *Autonomous Robots, Special Issue: Selected papers from Robotics: Science and Systems 2008*, 26(2–3):123–139, 2009.

[20] Z. Sun, G. Bebis, and R. Miller. On-road vehicle detection: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(5):694–711, 2006.

[21] C. Urmson, J. Anhalt, H. Bae, J. Bagnell, C. Baker, R. Bittner, T. Brown, M. Clark, M. Darms, D. Demitrish, J. Dolan, D. Duggins, D. Ferguson, T. Galatali, C. Geyer, M. Gittleman, S. Harbaugh, M. Hebert, T. Howard, S. Kolski, M. Likhachev, B. Litkouhi, A. Kelly, M. McNaughton, N. Miller, J. Nickolaou, K. Peterson, B. Pilnick, R. Rajkumar, P. Rybski, V. Sadekar, B. Salesky, Y. Seo, S. Singh, J. Snider, J. Struble, A. Stentz, M. Taylor, W. Whittaker, Z. Wolkowicki, W. Zhang, and J. Ziglar. Autonomous driving in urban environments: Boss and the Urban Challenge. *Journal of Field Robotics Special Issue on the 2007 DARPA Urban Challenge*, 25(1):425–466, 2008.

[22] T. Vu. *Vehicle Perception: Localization, Mapping with Detection, Classification and Tracking of Moving Objects*. PhD thesis, Institut National Polytechnique de Grenoble, Grenoble, France, September 2009.

[23] C. Wang. *Simultaneous Localization, Mapping and Moving Object Tracking*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA, April 2004.

[24] C. Wang, C. Thorpe, and A. Suppe. Ladar-based detection and tracking of moving objects from a ground vehicle at high speeds. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 416–421, 2003.

[25] S. Wender and K. Dietmayer. 3D vehicle detection using a laser scanner and a video camera. In *Proceedings of Sixth European Congress on Intelligent Transport Systems*, pages 105–112, 2007.

[26] L. Zhao and C. Thorpe. Qualitative and Quantitative Car Tracking from a Range Image Sequence. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 496–501, 1998.