Motion Estimation from Range Images in Dynamic Outdoor Scenes

Frank Moosmann[1], Thierry Fraichard[2]

***Abstract*—Object-class independent motion estimation from range data is a challenging task. We present here a novel approach that is able to derive a dense motion field based on range images only. We propose to first segment the range image into segments using a recently proposed segmentation criterion. Motion is then estimated segment-wise with full 6 degrees of freedom. To that end, we introduce dynamic mapping, *i.e.* the accumulation of measurements for moving objects. We show experimentally that the approach is able to deliver a dense motion field which can then be used for object-class independent trajectory estimation.**

I. INTRODUCTION

In recent years the development of time-of-flight cameras and multi-layer laser scanners advanced rapidly. These sensors provide denser measurements than single-line laser scanners and more precise distance measurements than stereo cameras, so new possibilities for object detection and tracking arise.

A common problem when working on laser scans or range images is that point correspondences cannot easily be established. In contrast to dense, texture-rich intensity images range data is sparser and local object geometries are not characteristic enough for matching purposes. Therefore, dense pixel-wise motion estimates do not yet exist. Most existing works on detection and tracking of objects in range data first apply some object model to detect and classify regions. Motion is then further estimated by applying data association to the detections at different times and by filtering the measurements using an object-specific motion model [1], [2], [3], [4]. While some approaches achieve impressive results even on sparse data, the major disadvantage is that these approaches only work for specifically modeled object classes.

Our goal is to develop an object-class-independent approach. In this work we present both a low-level segmentation cue and a low-level dense-motion-cue, similar to optical flow in intensity images. These allow for trajectory estimates and can later be used by higher-level algorithms to detect, classify, or track objects.

The proposed method works on range images, i.e. a two-dimensional array of distance measurements. Such images can be obtained directly from time-of-flight cameras or indirectly from multi-layer laser scanners. The latter take distance measurements sequentially which can be projected to a virtual image. We start by partitioning an image into maximally large segments. For each segment motion is estimated and filtered. To help data association, we introduce *dynamic mapping* – accumulation of measurements over time for static and moving segments. To keep the approach as general as possible, we work on the distance values alone and do not use any intensity, odometry, or positional information. We furthermore make no flat-world assumption so the approach should work in any type of terrain. We only assume rigid motion but show experimentally that the approach works even for articulated objects.

The paper is organized as follows: In the next section we position the proposed approach among existing works. We give an overview of the method in section III before we explain segmentation and motion estimation in more detail in sections IV and V respectively. Section VI shows some results before a conclusion and an outlook to future work is given in section VII.

II. PREVIOUS WORK

Approaches for object detection and tracking in range data are usually based on a ground plane assumption and combined with an occupancy grid map [5], [1]. Model-based object detectors are applied followed by a correspondence search between the models of subsequent frames. Such methods were applied successfully in the latest DARPA Urban Challenge (e.g. [2], [3], [4]). Surprisingly even vehicles equipped with multi-layer laser scanners projected all measurements to a ground plane which causes loss of information but enables the use of efficient, well-established 2D methods. Thus, all successful object tracking methods are 2D model-based, which requires manual model construction and model selection through classification.

Full 3D information is so far used solely by SLAM methods. Most of these methods only seek to estimate their own vehicle's motion and usually average out objects with different motion, as *e.g.* [6], [7]. As they work by aligning subsequent unordered point clouds, a high portion of moving objects (as occurs in heavy traffic) might cause these methods to fail. For lower outlier ratios, however, these registration methods provide good results. An overview of registration techniques can be found in [8]. Only few SLAM methods try to simultaneously detect and track moving objects, as *e.g.* [9]. Unfortunately, their computational efficiency and robustness for the application in 3D was not yet shown.

Interesting work that does not use object-class specific knowledge has been published in the domain of range images. Many approaches segment the image based on similar distance values followed by some tracking procedure, *e.g.* [10], [11]. Others segment the image by fitting planes to the range data. Sabata *et al.* [12] then group these planes and estimate motion

[1] Institute of Measurement and Control Systems, Karlsruhe Institute of Technology, Germany. frank.moosmann@kit.edu
[2] INRIA, CNRS-LIG & Grenoble University, France.

based on a graph representation. Altogether these methods follow a generic design but are still quite restricted in the application domain. They either work only for few special object classes or they require the objects to be well separated from background.

To the best of our knowledge, no approach exists yet that generically estimates a dense motion field for cluttered outdoor scenes. As a dense motion-field cannot be established robustly with single point correspondences, we propose to first segment the range image into parts by using a recently proposed local convexity criterion [13] and to estimate motion segment-wise using common registration techniques [8]. We further introduce *dynamic mapping*, accumulation of measurements for static and dynamic objects, which was inspired by the works of Gate *et al.* [14].

III. OVERVIEW

The proposed method works as illustrated in Fig. 1. The 2D range image domain is used for efficient segmentation, whereas motion estimation works on unordered 3D point clouds. These can be directly obtained from range images by using the physical sensor setup.

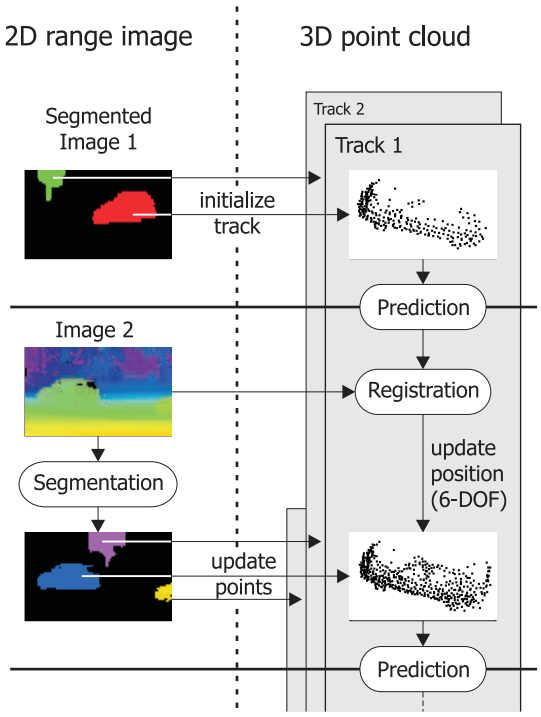

Fig. 1. Overview of the proposed method

A track consists of a state vector, which defines a local coordinate frame and a local appearance point cloud. A track is created whenever a segment is not assigned to an existing track. Otherwise the segment is used to update the tracks appearance by *dynamic mapping*. For estimating motion, the new range image is turned into a 3D point cloud, and each track is registered within this complete point cloud, independent from segmentation. A successful registration then causes an update of a track's state vector. The following sections describe the two main steps, segmentation and registration, in more detail.

IV. SEGMENTATION

Given a two-dimensional array of range measurements $R : (u, v) \mapsto r$, we explain here how to obtain large segments that are particularly suited for the following motion estimation. To increase readability, we subscript functions by index instead of using pixel coordinates as function arguments: $R(u, v)$ is thus denoted by R_i. Connections are implicitly established from each pixel to its 4 neighbors, also denoted by indices:

$$\begin{array}{ll} i_1 = (u+1, v) & i_3 = (u-1, v) \quad i_5 = i_1 \\ i_2 = (u, v-1) & i_4 = (u, v+1) \end{array}$$

Altogether, the following functions are used:

range measurement	R_i	:	$i \mapsto r$
euclidean coordinates	$\vec{E}_i$	:	$i \mapsto (x, y, z)^T$
distance vector	$\vec{D}_{i,j}$	=	$\vec{E}_j - \vec{E}_i$
connectiveness	$C_{i,j}$	:	$i, j \mapsto c$
normal vector	$\vec{N}_i$	:	$i \mapsto (n_x, n_y, n_z)^T$

Some of these relations are illustrated in Fig. 2 and their calculations are explained in the following.

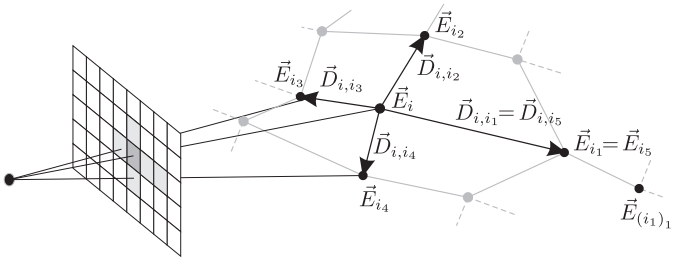

Fig. 2. Range image as implicit graph on 3D coordinates

The euclidean coordinates are directly obtained from the range measurements using the physical sensor setup. The distance vectors follow immediately. The connectiveness measure is a first indication for grouping pixels together and it is used to weight calculations on pixel connections. A pixel connection gets assigned a high connectiveness $C_{i,j}$ if neighboring distance vectors have similar length. As example, the connectiveness of a pixel to its right neighbor is calculated as

$$C_{i,i_1} = \min(\text{sigm}(|\tfrac{(R_i - R_{i_1}) - (R_{i_3} - R_i)}{(R_{i_3} - R_i)}|, \theta_1, c_1), \\ \text{sigm}(|\tfrac{(R_i - R_{i_1}) - (R_{i_1} - R_{(i_1)_1})}{(R_{i_1} - R_{(i_1)_1})}|, \theta_1, c_1))$$

The following sigmoid-like function serves as soft threshold

$$\text{sigm}(x, \theta, c) = 0.5 - \frac{0.5(x - \theta)c}{\sqrt{1 + (x - \theta)^2 c^2}}$$

where θ specifies the effective threshold and c a constant scale parameter to influence the tangent inclination at the threshold.

Second, a local surface plane represented by its normal vector is estimated at each measurement. For a given pixel with its 4 neighbors the normal vector is calculated as the

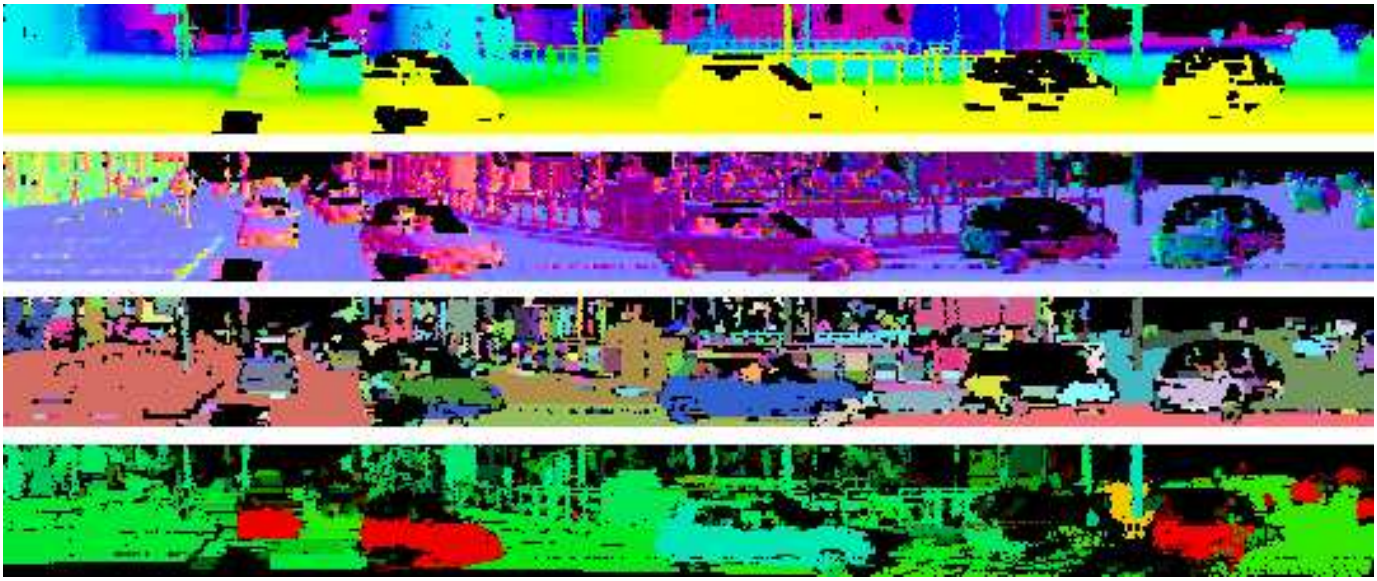

Fig. 3. Visualization of the involved steps: *Top row*: range image colored by distance magnitude, black pixels indicate missing measurements. *Second row*: estimated normal vectors for each measurement colored by normal direction. *Third row*: segmentation result, each segment is displayed in a different color. Tiny segments were removed, as motion cannot be estimated well enough. *Bottom row*: motion estimates, colored by magnitude of resulting 3D translation vector. For better visualization, images were cut on the left and odometry was used to compensate ego-motion.

average of the 4 cross products, each weighted by the product of their connectiveness:

$$\vec{N}'_i = \sum_{j=1}^{4} C_{i,i_j} C_{i,i_{j+1}} (\vec{D}_{i,i_j} \times \vec{D}_{i,i_{j+1}})$$

A moving average filter is then applied to the field of surface normals in order to reduce noise:

$$\vec{N}_i = \frac{\sum_{j=1}^{4} \vec{N}'_{i_j}}{||\sum_{j=1}^{4} \vec{N}'_{i_j}||}$$

Finally, segmentation is carried out. The method builds upon the *Local Convexity* criterion which was introduced in [13]. The idea is that many object parts have a convex outline, so surfaces are grouped together if they are locally convex to each other. In contrast, every border between an object and (flat) ground is concave, so these objects are never grouped together. Here, we improve its robustness by adding a twisting-constraint, by using the *connectiveness* values, and by applying fuzzy logic. The above defined sigmoid-like soft threshold replaces the hard thresholds used in [13]. Two neighboring pixels i, j connect if $C_{i,j} \cdot L_{i,j} \geq 0.5$, where $C_{i,j}$ is the previously described connectiveness and

$$\begin{aligned} L_{i,j} = \max\{\ & \text{sigm}[\vec{N}_i \cdot \vec{N}_j, 1 - ||\vec{D}_{i,j}|| \cdot \cos(\tfrac{\pi}{2} - \epsilon_1), c_2], \\ & \min[\text{sigm}(\vec{N}_j \cdot \vec{D}_{j,i}, ||\vec{D}_{j,i}|| \cdot \cos(\tfrac{\pi}{2} - \epsilon_2), c_2), \\ & \text{sigm}(\vec{N}_i \cdot \vec{D}_{i,j}, ||\vec{D}_{i,j}|| \cdot \cos(\tfrac{\pi}{2} - \epsilon_2), c_2), \\ & \text{sigm}((\vec{N}_i \times \vec{D}_{j,i}) \times \vec{N}_j, ||\vec{D}_{i,j}|| \cdot 0.3, c_2), \\ & \text{sigm}((\vec{N}_j \times \vec{D}_{i,j}) \times \vec{N}_i, ||\vec{D}_{i,j}|| \cdot 0.3, c_2)]\} \end{aligned}$$

is the modified *Local Convexity* criterion. The first term gives a value close to 1 if the two normal vectors are similar, the next two lines give a value close to 1 if each measurement is beneath the other's surface. The last two lines prevent the connection of twisting surfaces. ϵ_1, ϵ_2 define threshold angles, c_1 the thresholds' tangent inclination.

Segmentation is carried out using region growing with random seeds. As the segmentation criterion is symmetric, the outcome of the algorithm is nevertheless deterministic. The complexity of $O(\#\text{pixels})$ makes the method very fast.

Fig. 3 shows an example range image, the estimated normal vectors, and the resulting segmentation. As the proposed approach results in more complex segments than planes, motion can be estimated with full 6 degrees of freedom (DOF), as explained in the next section. Tiny segments are removed, however, as they cannot serve for good motion estimates.

V. MOTION ESTIMATION

Given the segments from the previous section, we now seek to estimate their 6-DOF motion, *i.e.* translation and rotation with respect to the next frame. We achieve this by a combination of methods: Feature matching, ICP, Kalman filtering, and *dynamic mapping* – detailed in the following.

Motion estimation is carried out on the new frames 3D point cloud independently for each *track*. A track is defined by its state vector $\vec{x} = (x, y, z, \psi, \theta, \phi, \dot{x}, \dot{y}, \dot{z}, \dot{\psi}, \dot{\theta}, \dot{\phi})^T$. The first 6 entries define a local coordinate frame, the other 6 entries its 6-DOF velocity. A track further stores its *appearance* – an unordered point cloud in 3D coordinates *wrt.* the track's local coordinate frame. Additionally, both the state vector and the appearance have associated uncertainties: A 12x12 covariance matrix for the state vector and a 3x3 covariance matrix for each point in the appearance point cloud. Fig. 5 depicts an example track at different points in time. All black appearance points are stored relative to the local (colored) coordinate frame.

In the first frame, each segment is turned into a track. The first 6 scalar state variables can be chosen arbitrarily, as they define some local coordinate frame. We use the absolute position of one of its measurement points and 0 for all orientations. The velocity part of the state vector is initialized with 0 but characterized by a high covariance. The appearance point cloud is initialized by adding all pixels of the segment as 3D points *wrt.* the track's local coordinate frame and associating the measurement noise as covariance.

Given a new range image, the motion of each track is estimated. A 6-DOF transformation is searched for that minimizes the sum of squared errors between each track's appearance point and its closest correspondence of all new measurements. This step is divided into an initial feature-based estimation step and a refinement step afterwards.

A. Initial Estimation

The Kalman filter is used to get an initial prediction by applying a constant velocity model. For each appearance-point of the track, a correspondence search is executed as explained in section V-B. If the average distance is above a specified threshold, feature matching is applied to get a better initial estimate.

For each pixel in the new frame a feature vector $f_i = (\vec{E}_i^T, R_i, (R_i - R_{i_1}), (R_i - R_{i_2}), (R_i - R_{i_3}), (R_i - R_{i_4}))^T$ is built and all features are organized within a kd-tree. Each pixel in the old frame then gets assigned its closest neighbor in feature space by searching the kd-tree. Due to the choice of the feature vector, both Cartesian coordinates as well as the local shape influence the search results. More sophisticated descriptors (see [8] for an overview) could be used as well, however, the possible improvement in matching would not justify the increased processing time.

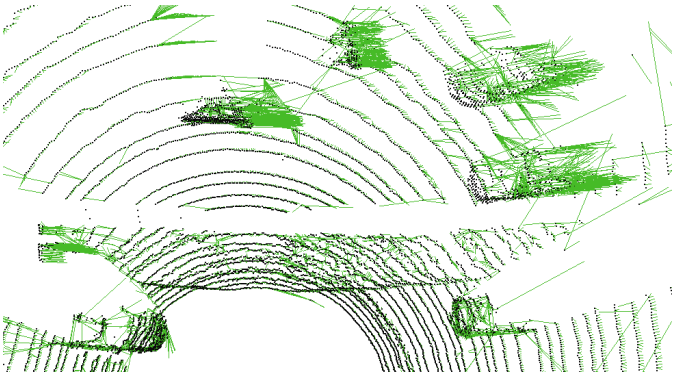

Fig. 4. Initial estimation by feature matching (bird's eye view): black points visualize measurements of the current frame, green lines indicate point matches between current and last frame

Fig. 4 shows an example result of the feature matching. It can be seen that estimates are quite noisy which prohibits their use as pixel-wise motion. Yet, if the translations given by the matches are averaged out for each segment, they serve as good initial motion estimates which are then further refined, as explained next.

B. Refinement

To refine a given estimate the ICP algorithm is used [15]. Iteratively, correspondences between each track's appearance point and the whole new point cloud are searched and distances minimized. Correspondence search is carried out on 3D-cartesian coordinates using a kd-tree. As normal vectors for each measurement are available, we employ minimization of the point-to-plane error of [15]. Iteration is continued for a maximum number of times or until the average correspondence error falls below a given threshold.

The refined motion estimate is then used for a Kalman filter update on the track's state vector. This will move a track's local coordinate frame and along with it the track's appearance point cloud.

C. Dynamic Mapping

After refined motion estimates are obtained, the appearance of each track is updated. In contrast to standard mapping techniques, which are either applied to static scenes or combined with an occupancy grid in order to average out measurements on moving objects, we refer to *dynamic mapping* as an approach which tries to accumulate appearance details of both static and dynamic objects.

This is achieved by first projecting the appearance points of all tracks to the current image. Then, the segmentation procedure (section IV) is executed and connections are established between segments and projected tracks. Segments that overlay a specific track to a high proportion are used for dynamic mapping as explained in the following. Segments that are not associated with any track are turned into new tracks, as explained at the beginning of section V.

To update the *appearance* of a track, all 3D points of the connected segments are added *wrt.* the track's local coordinate frame. In order not to accumulate an infinite number of points over time, the measurement and position uncertainty which are stored with each measurement and with the state vector are used to manage the local point cloud. A new point p_i with covariance matrix Σ_i is accepted, if the Mahalanobis distance $d_{\Sigma_i}(p_i, p_j)$ to any existing point p_j exceeds 1. If accepted, all existing points p_j with associated covariance matrix Σ_j are removed[1], if $d_{\Sigma_j}(p_i, p_j) \leq 1$.

Fig. 5 and Fig. 6 show typical results of dynamic mapping.

VI. EXPERIMENTS

We carried out experiments on data collected from a Velodyne HDL-64 laser scanner, mounted on top of our experimental vehicle. This scanner delivers 64 lines of measurements in a complete 360° view at 10Hz. We project these measurements to a virtual range image with a resolution of 870x64 pixel.

Fig. 3 shows an example image in an urban setting. Invalid measurements occur frequently, mainly in the sky and on close-by cars. The cross-product based normal vector estimation delivers accurate results, though some noise is clearly

[1]Thus, a new measurement replaces existing ones, if they are very close to each other and if the new measurement has less uncertainty

Fig. 5. Dynamic mapping: result of iterative motion estimation and accumulation of measurements when passing by a pedestrian (with attached local coordinate frame). Non-rigid objects will be mapped with more noise but data association still benefits from dynamic mapping

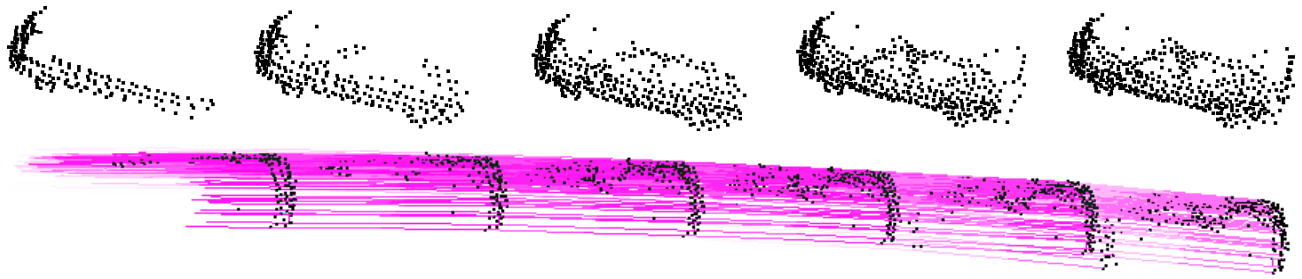

Fig. 6. Dynamic mapping: result of iterative motion estimation and accumulation of measurements when passing by a car. *Top row*: accumulated measurements of five selected frames viewed from the side. *Bottom row*: overlayed motion estimates of every second frame in a bird's eye view perspective

visible. The segmentation method slightly over-segments the scene which is preferable to under-segmentation: as motion is estimated for each segment, under-segmentation could result in wrong motion estimation whereas over-segmentation leads only to more noise in the estimation process. However, tiny segments ($<$5 pixel) have to be removed completely, as motion cannot be estimated reliably enough. In contrast to existing generic segmentation methods, the proposed method returns segments that are more complex than planes. This is extremely useful for motion estimation, as illustrated in the last row, which can thus be carried out with full 6-DOF. Further motion estimates can be seen in Fig. 7 and 8.

Results for *dynamic mapping* are depicted in Fig. 5 and 6. Motion estimation benefits thereof as soon as measurement resolution decreases, objects get occluded, or measurement failures occur (*e.g.* on close-by cars). Some of these situations are depicted in Fig. 7. Close to the sensor (regions 3 and 4) only few differences can be seen: Both the pedestrian on the right and the bicycle on the left are tracked well. As the resolution is high, data association is straight-forward so motion estimation can be performed even on small segments without dynamic mapping. In the background, however, the differences become clearly visible. In region 1 dynamic mapping could bridge the measurement gap for both cars. Without mapping wrong data association occurred, so the second car moved "into" the closer one. In region 2 dynamic mapping again successfully helped to track the vehicle, whereas without dynamic mapping the track got lost.

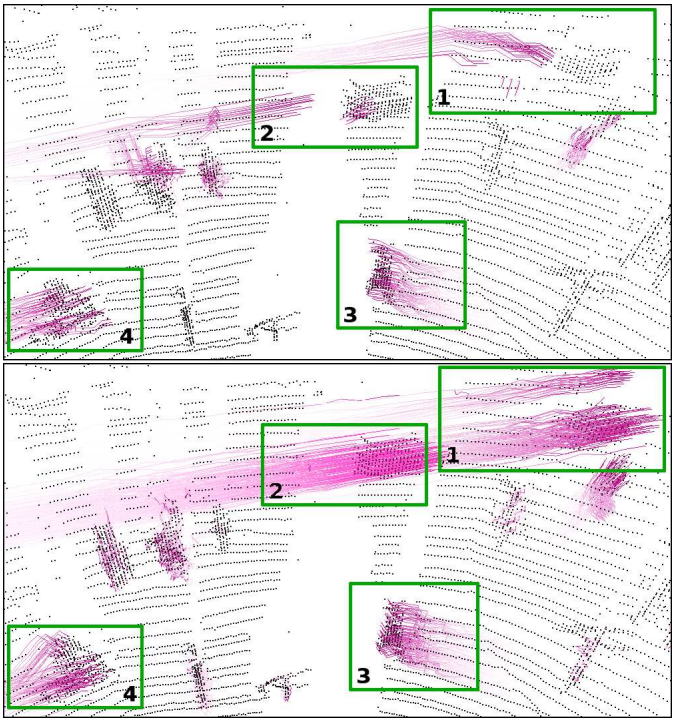

Fig. 7. Motion estimation over 30 frames at an intersection, without (upper image) and with (lower image) dynamic mapping. Black: measurements of 30th frame, purple: most prominent motion estimates over 30 frames. The lower object in region 1 was successfully tracked in both cases, its motion was suppressed in the lower image to better highlight the other objects motion

Fig. 9 depicts the covariance magnitudes of the Kalman filter for the lower object in region 1 of Fig. 7 over all 30 frames. Dynamic mapping leads to less uncertainty in the tracking process, which clearly helps to get robust dense motion estimates. Note that thereby, as argued in section V, the number of accumulated points always converges, which makes the method computationally tractable.

Additional material and results are available through our website at http://www.mrt.uni-karlsruhe.de/z/publ/download/rangeimagemotion/

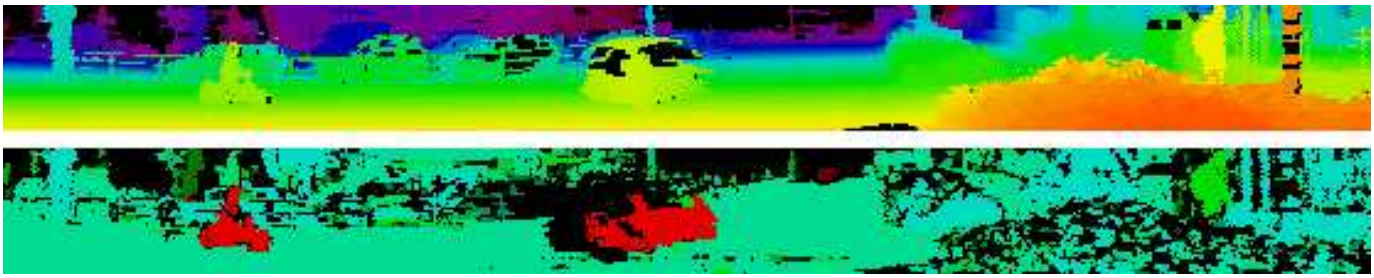

Fig. 8. Further results: *Top row*: range image colored by distance magnitude. *Bottom row*: motion estimates colored by magnitude of resulting 3D translation vector. For better visualization odometry was used to compensate ego-motion. Note that even the distant car in the upper middle and the partly hidden pedestrian on the right were correctly tracked.

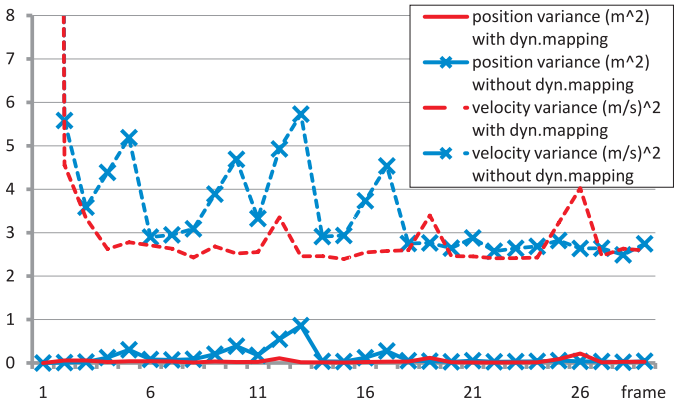

Fig. 9. Covariances taken from the Kalman filter over 30 frames

VII. CONCLUSIONS AND FUTURE WORK

We presented a novel method to get dense motion estimates in range images. Unlike the global approach of first estimating ego-motion by doing full scan-wise registration and afterwards determining inconsistent regions, the proposed approach can both handle higher outlier rates and segment even non-moving objects for a possible subsequent classification step. The image is first decomposed into segments and motion is estimated segment-wise. We improved a recently proposed segmentation method that produces segments with local surfaces in various directions. In contrast to plane-only segments, this allows the estimation of full 6-DOF motion. For motion estimation, we further proposed *dynamic mapping*, *i.e.* segment-wise accumulation of appearance, to help data association and thus motion estimation in low-resolution areas. Finally, we evaluated the proposed method on data collected in an urban setting. The method was able to estimate motion for most segments, resulting in a dense motion field.

Our next steps will include to move on from low-level segmentation and motion estimation to object detection. The idea is to follow the object-class-independent approach and to introduce a track merging/splitting framework grouping similar motion in order to obtain reliable trajectory estimates.

VIII. ACKNOWLEDGMENTS

The authors gratefully acknowledge the contribution of the German collaborative research center on Cognitive Automobiles (SFB/Tr28) granted by Deutsche Forschungsgemeinschaft as well as the support from Karlsruhe House of Young Scientists.

REFERENCES

[1] T.-D. Vu, J. Burlet, and O. Aycard, "Grid-based localization and online mapping with moving objects detection and tracking: new results," in *IEEE Intelligent Vehicles Symposium*, June 2008, pp. 684–689.

[2] M. Montemerlo *et al.*, "Junior: The Stanford entry in the Urban Challenge," *Journal of Field Robotics*, vol. 25, no. 9, pp. 569 – 597, 2008.

[3] C. Urmson *et al.*, "Autonomous driving in urban environments: Boss and the urban challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.

[4] S. Kammel *et al.*, "Team AnnieWAY's autonomous system for the 2007 DARPA Urban Challenge," *Journal of Field Robotics*, vol. 25, no. 9, pp. 615 – 639, 2008.

[5] S. Thrun, "Learning occupancy grid maps with forward sensor models," *Autonomous Robots*, vol. 15, no. 2, pp. 111–127, 2003.

[6] M. Bosse and R. Zlot, "Continuous 3d scan-matching with a spinning 2d laser," in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE. Kobe, Japan: IEEE Robotics and Automation Society, May 2009, pp. 4312–4319.

[7] A. Nüchter, K. Lingemann, J. Hertzberg, and H. Surmann, "6D SLAM—3D mapping outdoor environments: Research articles," *Journal of Field Robotics*, vol. 24, no. 8-9, pp. 699–722, 2007.

[8] J. Salvi, C. Matabosch, D. Fofi, and J. Forest, "A review of recent range image registration methods with accuracy evaluation," *Image Vision Comput.*, vol. 25, no. 5, pp. 578–596, 2007.

[9] C.-C. Wang, "Simultaneous localization, mapping and moving object tracking," Ph.D. dissertation, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, April 2004.

[10] T. Schamm, J. M. Zollner, S. Vacek, J. Schroder, and R. Dillmann, "Obstacle detection with a photonic mixing device-camera in autonomous vehicles," *Int. J. Intell. Syst. Technol. Appl.*, vol. 5, no. 3/4, pp. 315–324, 2008.

[11] B. Fardi, J. Dousa, G. Wanielik, B. Elias, and A. Barke, "Obstacle detection and pedestrian recognition using a 3D PMD camera," in *IEEE Intelligent Vehicles Symposium*, 2006, pp. 225–230.

[12] B. Sabata and J. K. Aggarwal, "Surface correspondence and motion computation from a pair of range images," *Computer Vision and Image Understanding*, vol. 63, no. 2, pp. 232–250, 1996.

[13] F. Moosmann, O. Pink, and C. Stiller, "Segmentation of 3D lidar data in non-flat urban environments using a local convexity criterion," in *IEEE Intelligent Vehicles Symposium*, June 2009, pp. 215–220.

[14] G. Gate and F. Nashashibi, "Using targets appearance to improve pedestrian classification with a laser scanner," *IEEE Intelligent Vehicles Symposium*, pp. 571–576, June 2008.

[15] Y. Chen and G. Medioni, "Object modeling by registration of multiple range images," in *IEEE International Conference on Robotics and Automation (ICRA)*, April 1991, pp. 2724 – 2729.