Detection and motion estimation of moving objects based on 3D-Warping

Robert Kastner, Tobias Kühnl, Jannik Fritsch, Christian Goerick

2011

Preprint:

This is an accepted article published in IEEE Intelligent Vehicles Symposium (IV). The final authenticated version is available online at: https://doi.org/[DOI not available]

Detection and motion estimation of moving objects based on 3D-Warping

Robert Kastner[1], Tobias Kühnl[2], Jannik Fritsch[3] and Christian Goerick[3]

***Abstract*— The detection of objects in the surrounding environment is a key functionality for every Advanced Driver Assistance System (ADAS). Otherwise, it is impossible to realize any kind of advanced assistance functionality as e.g. a Forward Collision Warning. Besides static obstacles that limit the drivable area, dynamic objects can even be more dangerous due to their ego-movement. Therefore, a detection and also motion estimation of moving objects in the surrounding is essential for every ADAS. In this contribution we present a novel way to detect moving objects only with 3D-measurements of the current surrounding environment that we called 3D-Warping. But we are not only able to detect but also to provide the direction and speed of moving objects in world coordinates. The method is generic and can be applied on any kind of depth sensor, as e.g. a stereo-camera, time-of-flight, and 3D-laser scanner.**

Keywords: moving object detection, object motion estimation, driver assistance

I. Introduction

Typical state-of-the-art Driver Assistance Systems (DAS), which are available on the market, rely on radar for the detection of dynamic objects. Therefore, providing distance and speed of vehicles in the surrounding environment, e.g. used for an Adaptive Cruise Control (ACC). Nevertheless, for future Advanced Driver Assistance Systems (ADAS) additional information might be necessary, as e.g. an initial segmentation of the moving object, support for all kinds of moving objects and a detection independent of moving or static obstacles. Furthermore, in technical terms a higher resolution to ease the separation between objects and also a larger opening angle of the sensor would be performance-improving features.

Currently, few approaches exist that do not rely on radar sensors to detect ego-moving objects. In the majority of cases these approaches use a direct measurement of the motion (as e.g. an optical flow computation). Also a few approaches exist that use an indirect measurement by the generation of an environment model, as shown in the related work section.

In this contribution, we present a novel approach for the detection of moving objects based on 3D-measurements of the surrounding environment, the 3D-Warping: First, objects in 3D are segmented to build a 3D environment representation. Afterwards, the intersection (and additional features) are evaluated between different time-steps of the found segments to identify the same object in different measurements. Additionally, the ego-motion is compensated by a 3D warping of segments from previous time-steps. Finally, a residuum computation between the same segments of different time-steps allows the detection of moving objects. The computation of the residuum is done in 3D, thereby also providing the direction and amplitude of a moving object. Due to the generic character of the proposed 3D segmentation all types of objects can be extracted (bicyclists, cars, motorcyclists, trucks). The 3D measurements are carried out with a time-of-flight sensor (Photonic Mixer Device - PMD).

II. Related Work

In general, a number of publications deal with the detection of moving objects from a static platform and also with the detection of general obstacles, which is not the focus of this contribution. We focus on the detection of moving objects from a moving platform. Due to our ego-motion, simple approaches like difference images can not be applied.

In the approach of Bertozzi et. al. [1] radar and camera data are fused for the detection of moving obstacles. The image is used to provide precise boundaries of the radar-detected objects. Additionally, the authors propose a method called 'Motion Stereo' which uses the transformation to the birds-eye-view (BEV) to build an environment model. Since, the BEV is a metric representation the ego movement can be easily compensated by a translation and rotation. In summary, to extract dynamic regions from the BEV two consecutive metric images are subtracted and the residuum is transformed to the perspective image for the indication of moving objects. Also, additional algorithms as a pedestrian recognition are fused with the 'Motion Stereo'. However, a radar is still utilized in this approach.

Miyasaka et. al. [2] build a 3D occupancy grid by using a multi-layer LIDAR. First, they extract the ego-motion by measuring the translation and rotation between the (static) elements of two consecutive scans. Afterwards, all outliers of the measurement indicate possible moving objects. These are clustered to get objects and also tracked over time. The identified moving objects provide position, direction and speed according to the authors. Nevertheless, the used sensor provides only a few scanning layers resulting in a small vertical opening angle.

Wang et. al. [3], [4] describe a simultaneous localization and mapping (SLAM) approach, also capable of detecting

[1]The work was carried out during the PhD studies of the author in cooperation between Honda Research Institute Europe GmbH and Darmstadt University of Technology, the author now works for Honda R&D Europe (Deutschland) GmbH, D-63073 Offenbach, Germany. robert_kastner@de.hrdeu.com

[2]Darmstadt University of Technology, Institute for Automatic Control, D-64283 Darmstadt, Germany. The author now works for CoR-Lab, Bielefeld University, Germany.

[3]Honda Research Institute Europe GmbH, D-63073 Offenbach, Germany. {jannik.fritsch, christian.goerick}@honda-ri.de

and tracking moving objects (DATMO). Based on laser scanners they build an environment map with their SLAM algorithm that also provides a pose estimation. Similar to the previous approach, outliers of the mapping algorithm indicate possible moving objects. For the tracking and prediction of each group of outliers (moving object) they use the Interacting Multiple Model (IMM) estimation algorithm. While [5] is based on a similar approach, the authors propose an extension that relies on a probabilistic method for the differentiation between static and moving objects.

Another approach is described by Nguyen et. al. [6], where an elevation map decodes the occupancy of the surrounding environment. To this end, areas with a large elevation are occupied. Based on that, areas whose occupancy status changes between current and previous measurements are likely to contain dynamic objects. To differentiate between false-positive measurements and real dynamic objects fuzzy-sets for a number of features are used, providing a probability for a dynamic object.

Summarizing, most of the approaches that use an environment representation also adopt a laser scanner, only providing precise information on a few planes.

One of the early approaches in the vehicle domain based on optical-flow is described by Heisele et. al. [7], where an image is divided by a simple color segmentation into 16 predefined levels. After the segmentation of connected regions, a number of features are extracted. The moving objects are determined by a search for corresponding regions between different time-steps and result in a motion image.

Wang et. al. [8] use stereovision and an approach called 'U-V-disparity' to extract 3D motion vectors, based on a segmentation of the disparity image and a comparison with the ego-motion.

A so called '6D vision' approach (see [9]) is used by Rabe et. al. [10] to detect moving objects. Therefore, the 3D positions from a stereo camera are combined with optical-flow vectors and the found objects are tracked with Kalman-filters. But the detection is only done for a number of feature points that were selected by a modified KLT-tracker (see [11]) from the original image. The detection of a moving object is realized with the Kalman-filter, that predicts the speed of an object and all objects with a speed greater than zero are moving.

The publication of [12] compares the previously introduced '6D vision' with a monocular motion detection. The monocular approach also uses a number of selected feature points (by the KLT-tracker) to compute the optical flow. The results showed that the stereo approach outperformed the monocular one in terms of accuracy.

Another approach is described by Fardi et. al. [13], also using the optical-flow for segmentation and classification of the object motion. Therefore, they use the RANSAC (see [14]) algorithm to differentiate between static and dynamic points. The static points are used to determine the ego-motion of the camera. Afterwards, the dynamic points are extracted with the knowledge of the camera ego-motion.

Optical-flow based approaches have in general two major drawbacks. First, the optical-flow computation is resource demanding. Second, the image-plane is ideal to detect motion lateral to the own movement direction, but has difficulties with the longitudinal direction. Additionally, the quantization error of stereo vision is a limiting factor for precise measurements.

III. SYSTEM DESCRIPTION

In the following, a rough overview of our approach for moving object detection is given (see Fig. 1). Thereafter, all processing steps and their theoretical background are described in more detail.

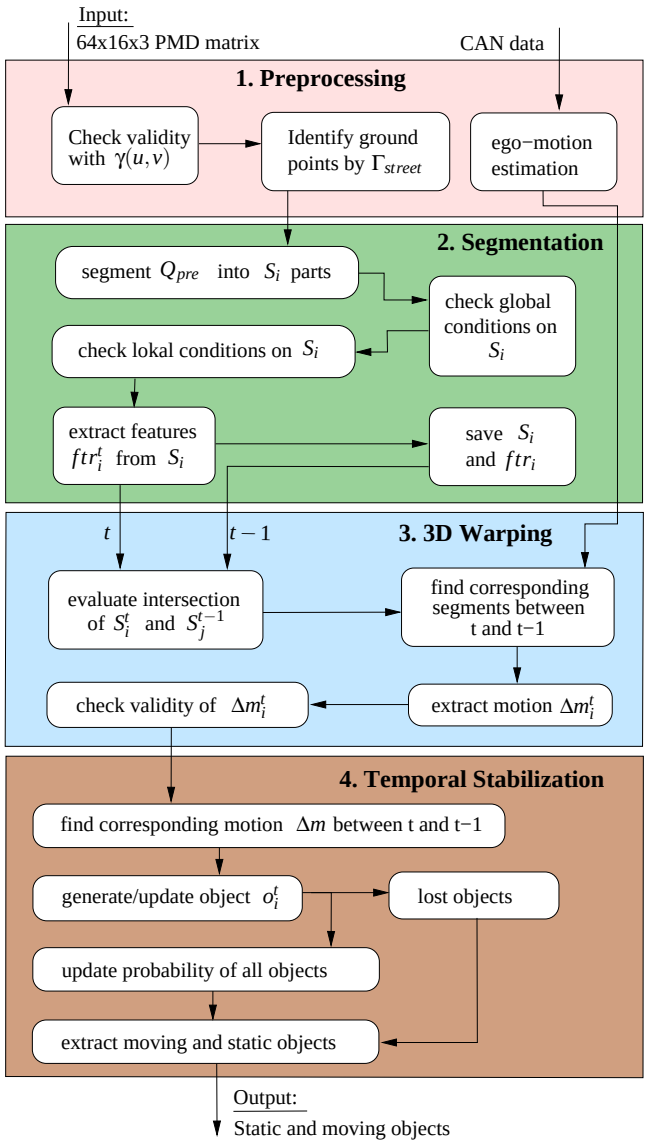

Fig. 1. System structure for detection of moving objects by 3D-Warping.

The used PMD sensor has a resolution of 64 x 16 measurement points and a framerate of 60Hz. Additionally, but only for evaluation purposes a camera is used. All sensors (PMD and camera) are mounted behind the windshield on the passenger's side of our test vehicle. The active infrared

illumination of the PMD is positioned at the radiator cowling of the test vehicle.

The overall system can be divided into four main parts. The first one handles the raw data preprocessing, where noise and systematically inaccurate measurements are eliminated by a validity check on the PMD data. The ego-movement of the test vehicle is calculated by using a single-track model on the speed and yaw-rate coming from inertial sensors. In the second part a segmentation is done to cluster the relevant segments of the PMD matrix Q. The found segments have to fulfill certain requirements which are global and local conditions corresponding to general and segment specific features/requirements. Afterwards, the features of the segments are extracted. In the following, third part the so-called 3D-Warping is applied. Starting with an intersection analysis to combine the same segments between the different time-steps t and $t-1$. Therefore, a number of features, e.g. size, width, height, additionally to the overlap criteria are compared to find the best match for each segment. Afterwards, the motion m_i^t for each segment S_i between the current time-step t and the previous one $t-1$ is extracted. Finally, in the fourth part a temporal stabilization is carried out, matching motions m with similar features between the different time-steps (at least three) to one object o_i. For the visualization of results the objects (in metric world coordinates) are projected to the image plane by a pin-hole camera model.

For the sake of simplicity, we switch between the matrix notation (e.g. A) and the function notation (respectively a(u,v)), nevertheless both notations are equivalent. Also, the superscript *t* for the current time-step is mostly omitted.

A. Preprocessing

The input of the system is a 64x16x3 PMD matrix Q ($q(u,v,w)$) for the current time-step t, having three layers which correspond to the X, Y and Z (depth) plane. Therefore, each element from a plane provides a measurement in metrical coordinates of the environment for a certain direction. A number of noisy or wrong measurements tend to happen, see Fig. 2. To this end, we filter Q by applying a validity interval Eq. (1), resulting in an Boolean matrix Γ_{val} for all valid points.

$$\gamma(u,v) = 1 \begin{cases} X_{min} \leq q(u,v,1) \leq X_{max} \\ Y_{min} \leq q(u,v,2) \leq Y_{max} \\ Z_{min} \leq q(u,v,3) \leq Z_{max} \end{cases}$$
$$\gamma(u,v) = 0 \text{ otherwise} \qquad (1)$$

Additionally, a number of measurements correspond to the street surface, which are insignificant for moving object detection. These measurements are identified by the Boolean matrix Γ_{street} generated by a street surface segmentation that starts at the lower bound of the Q matrix. The segmentation algorithm is the one from Section III-B, but modified to extract the ground plane. Therefore, the final validity matrix is given by Eq. (2) and the result of the preprocessing stage

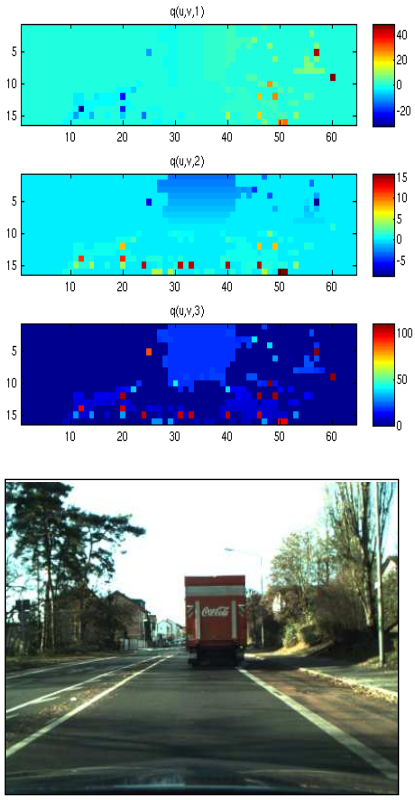

Fig. 2. Noisy raw data of the PMD sensor and image of the same time-step. Note that the camera and PMD sensor have different view angles, therefore a mapping is only possible by world positions and the use of a camera model.

by Eq. (3). $A \circ B$ is the entry-wise product of the matrices A and B.

$$\Gamma_{pre} = \Gamma_{val} \circ \overline{\Gamma_{street}} \qquad (2)$$

$$Q_{pre} = Q_w \circ \Gamma_{pre} \quad \forall\ w \qquad (3)$$

B. Segmentation

The second part of the algorithm handles the segmentation of relevant areas from Q_{pre}. The segmentation algorithm is inspired by basic region growing and split & merge algorithms (see [15], [8]). Also incorporated is the idea that measurements which belong to the same object lie within a certain depth interval. For the segmentation a number of start points (seed points) have to be defined. In general, there are two cases, first if the segmentation of the previous time-step $t-1$ exists, then a number of seed points for the current time-step t are provided. Second, without knowledge about a previous segmentation.

In the former case, all center points ($ftr_{l,1}^{t-1}$) of the segments S_l^{t-1} (S_i describes a segment i as binary matrix) are used as seed points for the current time-step t. The Moore neighborhood of the seed points is evaluated by use of the Euclidean distance metric $||a-b||_2$ (in metric coordinates). Therefore, the Euclidean distance between two

neighboring points must not exceed a certain threshold for being grouped to the same segment. After checking the neighborhood of the seed point the newly added segment points are evaluated in the same way. This procedure is repeated until no more neighboring points of a segment satisfy the distance constraint. In that case the method is continued with the next seed point.

All remaining points that have not been segmented by $ftr_{l,1}^{t-1}$ can act as new seed points. Hence, the Γ_{pre} matrix is searched row wise for remaining valid points acting as new seed points. If a new seed point is found the segmentation is started again as already described. After finishing a segment the row wise search of Γ_{pre} for remaining valid points is continued. This procedure is repeated until all valid points have been segmented. By using Γ_{pre} for the segmentation we simplify the retrieval of valid points, nevertheless the distance between two points is measured in 3D metrical coordinates.

Also, for the second case (without knowledge of previous segments) the Γ_{pre} matrix is searched row wise and found valid points act as new seed points.

The next step is a check for global and local conditions on all extracted segments S_i. The global conditions (GC) are on the one hand that a segment needs at least two points (GC_1, see Eq. (5)), otherwise it will be marked as invalid on Γ_{pre}. Since, single point measurements can not be separated from spurious detections.

$$n_i = \sum_u \sum_v s_i(u,v) \tag{4}$$

$$GC_1 : n_i \geq 2 \tag{5}$$

On the other hand, it is checked if a segment has two parts that are connected by a single point in between, assuming this is a false connection of two independent segments. Therefore, the connectivity of all valid points is calculated by Eq. (6).

$$\text{con}(u,v) = \sum_{n=u-1}^{u+1} \sum_{m=v-1}^{v+1} \gamma_{pre}(n,m) - 1 \tag{6}$$

Afterwards the continuity of the sums of each column is analyzed for each segment S_i (see Eq. (7)).

$$sc_i(u) = \sum_a (\text{CON} \circ S_i)(u,a) \tag{7}$$

If sc_i shows a discontinuity (or a number) the segment S_i is separated into several new segments K, which will replace the original segment i. This is the second global condition (GC_2).

The next step is the extraction of features $ftr_{i,j}$ for all segments S_i. To this end, for each segment i the center point in matrix ($ftr_{i,1} = [u,v]$) and metric coordinates ($ftr_{i,2} = [x,y,z]$), the size ($ftr_{i,3} = [width, height, length]$) and yaw angle ($ftr_{i,4}$) between segment and sensor is extracted. Based on these features a number of local conditions (LC) are evaluated for each segment S_i to reduce the amount of segments to only relevant ones. First of all, the pitch angle between sensor and segment must lie inside a validity interval, otherwise the segment is part of the street or the sky and will be marked as invalid. The second condition is the maximum length of a segment (see Eq. (9)), measured between the points with the minimum and the maximum depth. These segments belong to large man made structures like houses and walls, which are not of interest for the detection of ego moving objects.

$$Q_{i,l} = q_{pre}(u,v,l) \cdot s_i(u,v) \tag{8}$$

$$LC_2 : \max(Q_{i,3}) - \min(Q_{i,3}) \tag{9}$$

For the final third condition the standard deviation (see [16]) of the segments depth (Eq. (11)) or the standard deviation per point of a segment (Eq. (12)) has to be lower than a threshold. Otherwise, the segment is marked as invalid. The standard deviation per point is added for taking also the size of a segment into consideration.

$$\overline{z}_i = \frac{\sum_u \sum_v q_{i,3}(u,v)}{n_i} \tag{10}$$

$$\mu_{i,3} = \sqrt{\frac{1}{n_i - 1} \sum_{a=1}^{n_i} (q_{i,3}(a) - \overline{z_i})^2} \tag{11}$$

$$\frac{\mu_{i,3}}{n_i} \tag{12}$$

The result of the segmentation stage is therefore, S_i valid segments with $ftr_{i,j}$ features.

C. 3D Warping

The 3D Warping stage computes a motion m_i between the current time-step t with segment S_i^t and the previous time-step $t-1$ with a segment S_l^{t-1}. Hence, an intersection analysis for all current segments S_i^t with the previous segments S_l^{t-1} is carried out, resulting in three different cases:

1) there is no segment with an overlap in time-step $t-1$
2) there is exactly one overlapping segment of time-step $t-1$
3) there are a number of segments with an overlap from time-step $t-1$

The first two cases are simple, either the segment is stored for the next time-step because there is no match or there is exactly one match for further processing. In contrast to the third case, where additionally the size, number of points and depth difference between the segments are compared to find the best match for a segment S_i^t with S_l^{t-1}.

After finding two corresponding segments the motion between the time-steps is extracted by Eq. (13), where the ego motion of the car is compensated with Δm_{ego}^t.

$$\Delta m_i^t = ftr_{i,2}^t - ftr_{l,2}^{t-1} - \Delta m_{ego}^t \tag{13}$$

Therefore, the result of the 3D Warping stage is a motion vector Δm_i^t between two segments, describing the speed and direction in metrical world coordinates.

D. Temporal stabilization

After the extraction of a motion vector Δm_i between two segments as shown in the previous section, the results have to be stabilized over time and therefore assigned to certain objects. Based on the search for corresponding segments, also the correspondence between the motion vectors is realized. Hence, the motion vector Δm_l^{t-1} based on segments S_r^{t-2} and S_l^{t-1} is part of an object o_l. In the current time-step the motion vector Δm_i^t (between segments S_l^{t-1} and S_i^t) corresponds with the object o_l based on the same segment S_l^{t-1} of both motion vectors. Therefore, the speed and direction of o_l is averaged over the measurements Δm_l^{t-1} and Δm_i^t. Additionally, to the features of the measurements each object has a probability, which specifies if it is static or moving. To this end, an object can have three states:

1) new/unknown ($p_i = 0.5$)
2) static ($p_i < 0.5$)
3) moving ($p_i > 0.5$)

If a motion of a segment has no corresponding previous motion vector (and therefore no object in a previous time-step) a new object is generated with a probability (p_i) of 0.5 meaning the state is new/unknown. At each time-step the probability p_i of an object is updated based on the previous probability (p_i^{t-1}) and the similarity of the current direction ($p_{i,D}^t$) and amplitude ($p_{i,A}^t$) with the previous ones (see Eq. (14)). If the amplitude is below 3km/h $p_{i,D}^t$ and $p_{i,A}^t$ are zero.

$$p_i^t = p_i^{t-1} + p_{i,A}^t + p_{i,D}^t - \Delta p_{i,H} \tag{14}$$

Additionally, for small magnitudes of the amplitude ($< 3km/h$) a penalty factor $\Delta p_{i,H}$ exists reducing the overall probability. Therefore, the probability p_i indicates the certainty with which an object is moving or static.

IV. RESULTS

In this section, we evaluate the performance of our system with an image stream showing different scene categories. For the evaluation the found moving objects are transformed to the image plane by a pin-hole camera model. Additionally to the single segment points also the corners of each segment are transformed to the image plane (only for moving objects). The segment corners span a rectangle on the image plane, which will be further used for the evaluation. One image per second was manually labeled, providing the ground truth (GT) position and size of all moving objects (trucks, cars, motorcycles, bicyclists) on an image (img). In general, also pedestrians can be detected as moving objects, but the resolution of the PMD sensor is too small to generate enough measurement points in the required distance.

A true-positive match is given, if the mutual overlap of the ground truth region with the found moving object rectangle is greater than 50% (relating to the smaller region). The streams show various scenes, with different scene complexities (see Fig. 3). On the image plane, the evaluation measure treats each moving object independently. To this end, each of the moving objects on an image has to be classified.

In order to evaluate our algorithm, we adopt the Equations (15), (16), and (17). The equations define different ground truth based measures, which were taken from [17].

$$\text{Completeness} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{15}$$

$$\text{Correctness} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{16}$$

$$\text{Quality} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}} \tag{17}$$

with

TP ... True positive moving objects
FN ... False negative moving objects
FP ... False positive moving objects

On a descriptive level the Completeness states, based on given ground truth data, how many of the moving objects were actually detected. The Correctness states how many of the detected regions were actually relevant moving objects. The Quality combines both measures. Its computation is appropriate, since a trade-off between the Completeness and Correctness exist. Based on this, the Quality measure should be used for a comparison, since it weights the FP and FN detections equally. For a more detailed analysis the Completeness and Correctness state what exactly caused a difference in Quality.

The three measures were calculated on the detected moving objects over all ground-truth images of the stream. The gathered results are depicted in Tab. I and Tab. II.

Category	Number of GT frames	TP	FP	FN
Rural to city	82	97	11	19
Inner city	137	149	10	41
Highway	151	295	35	17

TABLE I
GENERAL INFORMATION ABOUT THE EVALUATION STREAM.

The computed measures show the reliability of the approach, given that a Quality of at least 74% is reached. In detail the Correctness is about 90% on all categories. The false-positive detections are caused mostly by the used commercial internal vehicle sensors that measure the ego-motion of the test vehicle. Especially, during curves the quality of the ego-motion estimation drops significantly, which can produce false positive detections. The Completeness shows a higher variation between the different categories, but still nearly 80% of all moving objects can be detected. Since, the evaluation is done on a single frame basis (one per second),

Category	time [s]	FP per GT img	Correctness	Completeness	Quality
Rural to city	82	0.13	89.8%	83.6%	76.3%
Inner city	141	0.07	93.7%	78.3%	74.5%
Highway	153	0.23	89.4%	94.6%	85.0%

TABLE II
RESULTS OF THE EVALUATION

it happens periodically that a 'new' moving object has not been detected yet. Because our approach needs at least three measurements before a moving object is identified as such (see Figure 3, Frame 6883). Also partly occluded objects that are only visible for short intervals (see Figure 3, Frame 4250), produce false-negative measurements.

Fig. 3. Example images of the results, red points depict moving objects, green static ones. The moving objects show additionally rectangles of the temporal stabilization, where the color intensity indicates the certainty for moving. The average speed (km/h, after temporal stabilization) for the detected moving objects from our approach is also depicted. Note: in Frame 5360, the moving object on the right hand side is a bicyclist.

V. Summary and Outlook

In this contribution, we presented a novel way for the detection and speed/direction estimation of moving objects, only based on 3D measurements. As the results showed the method reliably detects moving objects and also the object speed estimation resulted in reasonable outputs. Therefore, we plan a sensor fusion with vision to further exploit the gathered results for a highly flexible and robust system architecture. We currently port the described extensions from Matlab to C in order to integrate them in our existing online system [18] for evaluating them on our prototype vehicle.

References

[1] M. Bertozzi, L. Bombini, P. Cerri, P. Medici, P. Antonello, and M. Miglietta, "Obstacle detection and classification fusing radar and vision," in *Proc. IEEE Intelligent Vehicles Symposium*, 2008.

[2] T. Miyasaka, Y. Ohama, and Y. Ninomiya, "Ego-motion estimation and moving object tracking using multi-layer lidar," in *Proc. IEEE Intelligent Vehicles Symposium*, 2009.

[3] C.-C. Wang, C. Thorpe, and S. Thrun, "Online simultaneous localization and mapping with detection and tracking of moving objects: theory and results from a ground vehicle in crowded urban areas," in *Proc. IEEE Internat. Conf. on Robotics and Automation*, 2003.

[4] C.-C. Wang, C. Thorpe, and A. Suppe, "Ladar-based detection and tracking of moving objects from a ground vehicle at high speeds," in *Proc. IEEE Intelligent Vehicles Symposium*, 2003.

[5] L. Montesano, J. Minguez, and L. Montano, "Modeling the static and the dynamic parts of the environment to improve sensor-based navigation," in *Proc. IEEE Internat. Conf. on Robotics and Automation*, 2005.

[6] T.-N. Nguyen, M.-M. Meinecke, M. Tornow, and B. Michaelis, "Optimized grid-based environment perception in advanced driver assistance systems," in *Proc. IEEE Intelligent Vehicles Symposium*, 2009.

[7] B. Heisele and W. Ritter, "Obstacle detection based on color blob flow," in *Proc. IEEE Intelligent Vehicles Symposium*, 1995.

[8] J. Wang, Z. Hu, H. Lu, and K. Uchimura, "Motion detection in driving environment using u-v-disparity," in *ACCV (1)*, 2006.

[9] U. Franke, C. Rabe, H. Badino, and S. K. Gehrig, "6d-vision: Fusion of stereo and motion for robust environment perception," in *DAGM-Symposium*, 2005.

[10] C. Rabe, U. Franke, and S. Gehrig, "Fast detection of moving objects in complex scenarios," in *Proc. IEEE Intelligent Vehicles Symposium*, 2007.

[11] C. Tomasi and T. Kanade, "Detection and tracking of point features," Carnegie Mellon University, Tech. Rep., 1991.

[12] T. Vaudrey, A. Wedel, C. Rabe, J. Klappstein, and R. Klette, "Evaluation of moving object segmentation comparing 6d-vision and monocular motion constraints," in *IVCNZ08*, 2008.

[13] B. Fardi, T. John, and G. Wanielik, "Non-rigid-motion recognition using a moving mono camera," in *Proc. IEEE Intelligent Vehicles Symposium*, 2009.

[14] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," in *Comm. of the ACM*, vol. 24, 1981, pp. 381–395.

[15] R. Gonzalez and R. Woods, *Digital Image Processing*. Addison-Wesley Publishing Company, 1992.

[16] L. Fahrmeir, R. Künstler, I. Pigeot, and G. Tutz, *Statistik - Der Weg zur Datenanalyse*, 6th ed. Heidelberg, Berlin: Springer, 2007.

[17] P. Lombardi, M. Zanin, and S. Messelodi, "Unified stereovision for ground, road and obstacle detection," in *Proc. IEEE Intelligent Vehicles Symposium*, 2005.

[18] J. Fritsch, T. Michalke, A. Gepperth, S. Bone, F. Waibel, M. Kleinehagenbrock, J. Gayko, and C. Goerick, "Towards a human-like vision system for driver assistance," in *Proc. IEEE Intelligent Vehicles Symposium*, 2008.