Environment Estimation with Dynamic Grid Maps and Self-Localizing Tracklets

Andrei Vatavu[1], Nils Rexin[2], Simon Appel[2], Tobias Berling[3], Suresh Govindachar[1], Gunther Krehl[1], Janis Peukert[1], Manuel Schier[1], Oliver Schwindt[3], Jakob Siegel[1], Christos Zalidis[3], Timo Rehfeld[1], Dominik Nuss[1], Michael Maile[1], Sven Zimmermann[3], Klaus Dietmayer[2], Axel Gern[1]

***Abstract*—Dynamic environment representation is an important and demanding topic in the field of autonomous driving. One of the generic ways to estimate the surrounding world of an intelligent vehicle is to use dynamic grid maps. However, there are still several unsolved challenges in the grid-based tracking solutions like the ability to converge faster and providing a more efficient way to fuse multi-sensorial information. In this work, we address both of these challenges as a single probabilistic estimator. First, we treat the grid map estimation process as a multi-channel tracking mechanism. In particular, we use a particle filter based solution to integrate both the occupancy and semantic grids. Second, we adapt the idea of simultaneous grid cell tracking and object shape estimation into the grid map domain and propose "self-localizing tracklets", which are individual particle filter based estimators that are used for two main tasks: stabilizing the position estimation accuracy of dynamic cells with respect to the object boundary, and estimating a better object shape. The presented concepts offer an improved representation flexibility and a faster algorithm convergence.**

I. Introduction

Dynamic environment perception is one of the core functions of an autonomous vehicle. The ability to reliably detect the surrounding traffic area plays an important role in many components of a self-driving car such as path planning, behavior generation, collision avoidance or self-localization. The complexity of developing a robust solution to perceive the dynamic world comes from different challenges. The autonomous vehicle should be able to cope with various use cases where the surrounding environment is crowded and unpredictable, with multiple static and dynamic objects (cars, poles, pedestrians, walls, bicycles etc.). This might include traffic intersections, construction zones, or parking areas.

In order to cover the perception requirements in different complex scenarios, various types of sensors are employed. Typically, laser scanners are used to provide accurate position [4], [6], [15], [24], stereo-vision or surrounding cameras are employed to provide both range and semantic information [10], [16], [17], while radars are more suitable to detect motion [8]. As the technology evolves and the computational power increases, perception and tracking architectures achieve a more comprehensive understanding of environment through sensor information fusion [5], [9], [10]. However, one has to consider that the vehicle's sensor setup often changes. As soon as the market price allows it, new, better performing sensors can be added or can replace older ones. Sometimes this might affect the entire processing chain. Therefore one of the imposed challenges is to design a dynamic environment estimation component that is scalable (new sensors can be added) and flexible (the estimation solution could be easily decoupled or adapted to different combination of sensors). At the same time, the environment perception should be able to cope with the cases when a part of the sensor system fails, by generating the right model with the remaining sensors.

The existing algorithms aim to provide the environment perception at different abstraction layers, depending on the complexity of the surrounding world, by pointing to various modeling and tracking solutions. A considerable research work has been focused on detecting obstacles in traffic scenarios [2], [3]. Some approaches attempt to model objects at a higher abstraction level by using oriented boxes [6], or L-shape models [4]. These simple models are convenient to describe structured environments such as on-road vehicles in highway scenarios. As long as the traffic participants fit the model this approach works well. However, the box representation is not sufficient to describe more sophisticated and unpredictable infrastructure. In order to improve the robustness various algorithms try to find a trade-off between the representation flexibility and computational efficiency. For example, the dynamic objects are modelled as deforming and moving contours [18], [20], parametrized curves [19], individually tracked 3D points [21], boxes with adaptive size [6], point sets describing rigid objects [24], voxels [22] or dynamic stixels [17].

A different abstraction layer is the result of estimating the environment at an *intermediate* level – a level that is able to provide higher accuracy, and flexibility than the object representations and a lower processing time than dense 3D point-cloud tracking methods. A well-established intermediate representation approach is grid mapping. Grid maps discretize the surrounding world into grid cells and keep the evidence of properties like occupancy [1]. The first occupancy grid implementations applied a Bayesian estimation scheme to incorporate successive range measurements and assumed a stationary environment. Later, various techniques were proposed to model and track dynamic occupancy grids. Coue et al [26] associate random variables to each cell for estimating occupancy and velocity. In Dansescu et al. [11] a particle filter

[1]Mercedes Benz R&D North America, Sunnyvale, CA, US.
[2]Ulm University, Germany.
[3]Robert Bosch LLC, Palo Alto, CA, USA.
Primary contact: andrei.vatavu@daimler.com

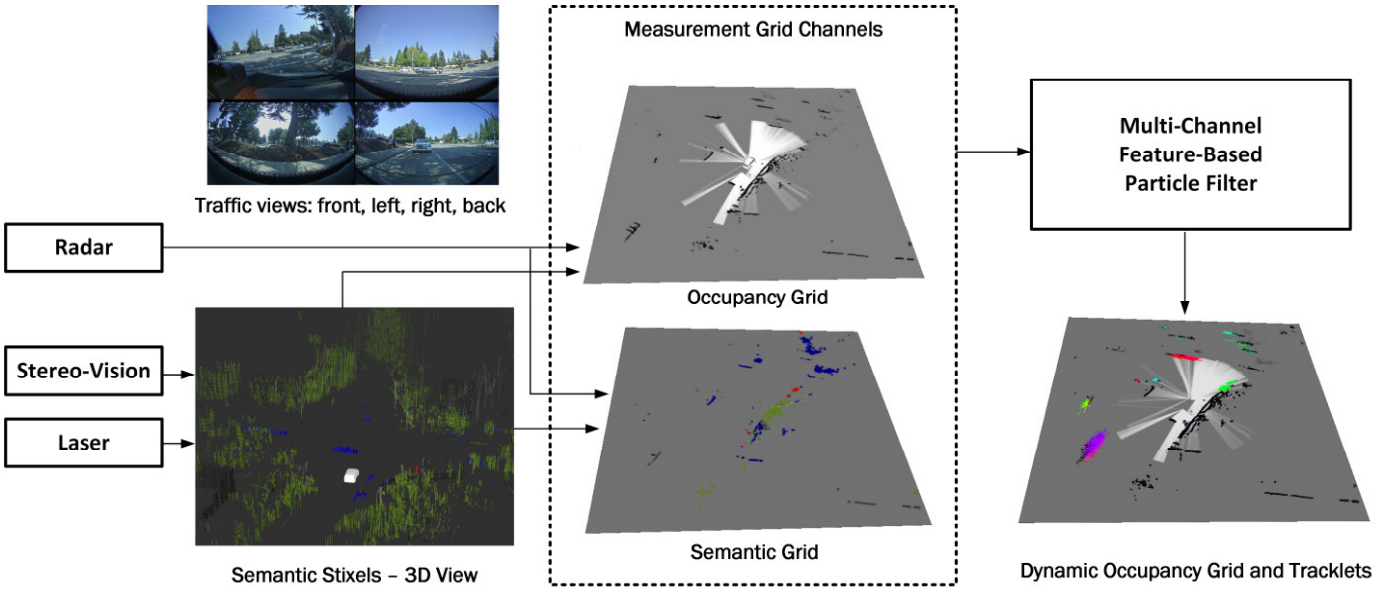

Figure 1. System Overview.

mechanism is used to estimate the grid cell occupancy and speed. The particles are considered to be independent hypotheses that have their own position and velocity. In [7], Nuss et al. present an approximation of the particle-filter based grid estimation, applying the Dempster-Shafer theory of evidence. In [13], [14] the particle filtering is only employed to estimate velocities. A mix of static and dynamic particles is applied in [14]. Early approaches were limited to estimating position and/or speed by projecting some form of range information (stereo, laser, radar) into the grid space and applying a Bayesian estimation. Recent approaches, however, improve the grid estimation by employing new measurement properties. For example in [12], Danescu et. al. extend the particle state with an additional dimension, the height. In [23], two grid maps (intensity and occupancy) are used to extract a set of rectangular 3x3 grid blocks. A Rao-Blackwellised solution is employed as a main estimator, where a particle model is extended with the intensity and occupancy information and is weighted based on how well it matches the extracted measurement blocks. Although the block-based approach is able to incorporate the appearance information, it is limited only to the close proximity of a particle.

In general, the grid-based tracking solutions have a common limitation. They are not able to estimate with high accuracy the state of cells belonging to large and uniform grid areas. For example, a particle predicted in the middle of a larger object can be assigned to any of the occupied cells, thus leading to higher uncertainties due to the ambiguous data association. In order to improve the estimation accuracy, for uniform grid areas (i.e. in the middle of large objects with the same occupancy values), we would need to use larger patches, employ extra features from various sensors, or increase the number of particles. Therefore, one of the questions that arises is: how to incorporate more knowledge about the surrounding world in the particle state, increase the flexibility and still keep a fixed memory space for each particle?

In this work we focus on extending the dynamic grid map estimation with two concepts. First, we treat the estimation process as a multi-channel tracking mechanism. A channel is a separate grid map that projects a group of sensors into a 2D space of discretized cells and has its own measurement model. Specifically, our estimation solution is based on using two channels: an occupancy grid and a semantic grid. Second, we introduce a particle-filter based estimator where the particle model is extended with partial knowledge about a given object shape, thus creating a bridge between low-level particle world and high-level object world. This concept helps us to "anchor" a given particle to a fixed sub-set of features pre-selected from a potential object hypothesis (grid blob). In other words, each particle has its own local map of object landmarks. These landmarks are used in the tracking process as an additional knowledge about the rigid object shape. Instead of keeping one big set of particles, we employ multiple smaller independent populations of samples organized into tracklets. Although, all particles from various tracklets share the same measurement space, they know the group (tracklet) they belong to. This strategy provides a higher flexibility:

- Some of the tracking operations (creation, deletion or update) can be performed by manipulating the particles in batches, at the tracklet level.
- A common grid cell property (e.g. semantic label, object ID etc.) is stored at the tracklet level and can be accessed by all its particles without the need of replicating the same data at the particle level. For example, all the particles could share the same semantic label stored as a tracklet property.
- Similarly to FastSLAM techniques [24] that use a set of measured landmarks to localize a robot in the map, the intermediate tracklet entities are self-localizing themselves with respect to an object boundary. Therefore we minimize the effect of "drifting tracklets" inside big and uniform objects due to the ambiguous particle-to-measurement associations. Although the extending particle model with landmarks inevitably increases the algorithm complexity, in the end we show that this is compensated by the fact that less particles are required to match the target.

We will refer to the proposed approach as **M**ulti-**L**ayer **P**article Filter-based **T**racking (MLPT) method. The rest of the paper is structured as follows: the overall system overview and processing pipeline is presented in the next chapter. Chapter III presents the concept of self-localizing tracklets and multi-channel grid estimation by using a grid-based particle filter mechanism. Chapter IV describes the main steps of MLPT. The experimental results are presented in the Chapter V whereas the conclusions about this work are described in the last chapter.

II. System Overview

This section provides an overview of our proposed approach and how the main system components are interconnected (see Fig. 1). The general processing flow can be decomposed into three main stages. First, the raw sensor information is transformed into more compact, intermediate data structures. The radar measurements are converted into a point set including both position and velocity properties. The radar sensors have similar description as in [28]. The stereo-vision images and LiDAR raw point clouds are both transformed into semantic Stixel elements. The Stixel abstraction model is a compact and rich medium-level representation in form of vertically oriented rectangles that incorporate both depth and semantic information [16], [17]. Besides the provided data structures, every sensor's input comes with its own pre-defined measurement model.

In the second stage of the processing flow, the medium-level representations are projected into evidence grid channels. To generalize the description, a channel can be computed by integrating multiple sensors and a sensor can contribute to multiple channels. In our specific case, we compute two measurement grids – an occupancy grid and a semantic grid. However, the proposed solution can be extended by adding new channels (e.g. heights, gradients etc.)

The measurement *occupancy grid* is obtained by integrating the range measurements from all the sensors accumulated during a fixed time interval. The measurements are combined into the grid space by applying the Dempster-Shafer theory of evidence where each cell is described by a mass of occupied and free. Its occupancy probability can be recovered by using the so called pignistic transformation. More details about the occupancy grid creation is provided in [7].

The measurement *semantic grid* integrates both the classified radar targets [28] and the Stixel world into the grid space. As a given object label (e.g. pedestrian, car, bicycle etc.) is associated with a confidence score [17], a semantic grid cell stores a maximum number of K object labels with the highest confidence.

Both grid channels are of the same size, and are aligned in space (a given grid cell index describes the same position in the environment), and time (the input measurements are synchronized and are collected into the grid space during a fixed time-slice). The last stage of the processing flow is the proposed hybrid particle filter based estimator which is the main focus of this work and is described in the next sections.

III. Dynamic Environment Estimation

In general, the main objective of a tracking process is to estimate the current state $\mathbf{s_t}$ of a target from a set of noisy

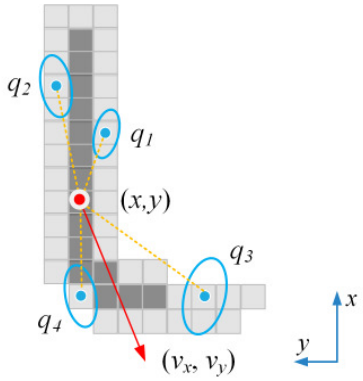

Figure 2. The tracklet model described by its position (*x*,*y*) and four reference object landmarks (blue) initiallized from the blob contour (light gray). The ellipses depict the landmark covariances. Although the example shows only one tracklet assigned to one grid cell (red), we initialize new tracklets with random landmark in every newly observed cell (a measurement cell that was not associated yet to any existing tracklets).

measurements $\mathbf{z_{1:t}}$ collected up to the time *t*. The estimation of the posterior probability distribution $p(\mathbf{s_t}|\mathbf{z_{1:t}})$ at time *t* can be formulated as a recursive Bayesian update rule using the probabilistic motion model $p(\mathbf{s_t}|\mathbf{s_{t-1}})$ of the target and a defined measurement model $p(\mathbf{z_t}|\mathbf{s_t})$:

$$p(\mathbf{s_t}|\mathbf{z_{1:t}}) = \eta p(\mathbf{z_t}|\mathbf{s_t}) \int p(\mathbf{s_t}|\mathbf{s_{t-1}}) p(\mathbf{s_{t-1}}|\mathbf{z_{1:t-1}}) ds_{t-1} \quad (1)$$

where η describes the normalization constant.

One specific implementation of the Bayes filter equation described above can be done by using a particle filter mechanism. At each moment in time *t*, the particle filter algorithm maintains a set of *N* samples $\{\langle \mathbf{s_t^{[i]}}, w_t^{[i]} \rangle\}_{i=1..N}$ to approximate the posterior $p(\mathbf{s_t}|\mathbf{z_{1:t}})$. Each particle $\mathbf{s_t^{[i]}}$ describes a guess of the object state and has an associated weight $w_t^{[i]}$.

A standard particle filter can be applied to multiple estimation problems involving non-linear motion or measurement processes. However, particle filters are restricted to estimate low dimensional states and are not suitable for representing larger states where the number of required samples to approximate the true belief might grow exponentially once more properties are incorporated into the state. As it is shown in some applications such as object tracking [6], [20], [23] or simultaneous localization and mapping [25], a common way to deal with larger states is to use a Rao-Blackwellised particle filter (RBPF) [27]. In a RBPF algorithm, the samples are drawn only from a part of the state (e.g. object motion), while the other state parameters are attached to each particle and are estimated in closed form. In order to improve the accuracy, we use the same Rao-Blackwellization process to introduce "richer" particles for modeling two main grid estimation parts that are combined into one single state estimator. The two parts are: *self-localizing tracklets* and *multi-channel grid estimation*.

A. Self-localizing tracklets

The majority of grid-based particle filters approximate the cell state with a set of samples. The particles are not assigned to a specific object hypothesis but rather can be propagated

into different grid cells according to their own motion model. As soon as the new target cells are sensed as occupied, the particles receive higher weights. In other words, the particles are weighted without being aware of their own position with respect to the tracked object hypothesis (see Fig. 3, left).

The central point of our proposed concept is to bring the idea of simultaneous grid cell tracking and object shape estimation into the grid map domain, at the cell level.

We consider that a given dynamic grid cell is part of a larger object hypothesis. Besides its position (x, y) and velocity (v_x, v_y) it is also represented by its relative position to K object landmarks $\mathbf{Q_t} = \{\mathbf{q_{t,1}}, \dots, \mathbf{q_{t,K}}\}$ initialized by randomly selecting a set of points from the object contour (see Fig. 2). In our case a landmark is defined as a 2D feature point belonging to a static or dynamic object contour. The landmark state is recursively updated based on the new observations, thus covering the change in blob-shape.

For every newly observed grid cell we create a fixed number N of particles. This group of particles will describe an independent particle-filter estimator – a tracklet. Thus, instead of keeping one big set of particles for the entire grid, we employ multiple smaller independent populations of particles organized into tracklets. In this context a tracklet position and velocity at time t will be denoted $\mathbf{x_t} = [x_t, y_t, v_{x,t}, v_{y,t}]^T$. Additionally the tracklet state will be described by a unique set of landmarks $\mathbf{Q_t}$ (see Fig. 2). In the end, different tracklets belonging to one grid blob will be represented by different combination of landmarks selected from the set of blob contour points. However the samples belonging to one tracklet will represent the hypotheses of the same unique set of landmarks used to define the tracklet state (all the particles will have more or less the same partial shape). The problem can be formulated probabilistically as estimating the joint posterior:

$$p(\mathbf{x_t}, \mathbf{Q_t} | \mathbf{z_{1:t}}) \tag{2}$$

Similarly to FastSLAM approaches [25] the problem can be implemented with Rao-Blackwellised particle filter and can be factored into independent estimators as:

$$p(\mathbf{x_t}, \mathbf{Q_t} | \mathbf{z_{1:t}}) = p(\mathbf{x_t} | \mathbf{z_{1:t}}) p(\mathbf{Q_t} | \mathbf{x_t}, \mathbf{z_{1:t}})$$

$$= p(\mathbf{x_t} | \mathbf{z_{1:t}}) \prod_{k=1}^{K} p(\mathbf{q_{t,k}} | \mathbf{x_t}, \mathbf{z_{1:t}}) \tag{3}$$

The main motivation of adopting this extension in the grid map space comes actually from the need of stabilizing the effect of "drifting" tracklets by improving the particle-to-measurement matching. More exactly, without shape information, which is modeled here as a set of anchor points, a tracklet would be more likely to be described by a higher uncertainty due to the limited ability of simple particles (described only by $\mathbf{x_t}$) to confirm the "right" measurements inside large and uniform blobs where every cell has the same occupancy value (see Fig. 3, left). By making the analogy with the SLAM techniques, we also could say that our tracklet models are able to self-localize themselves with respect to the object boundary (see Fig. 3, right). At the same time, this strategy allows us to create a bridge between low-level particle world and high-level object world by incorporating the partial knowledge about object shape.

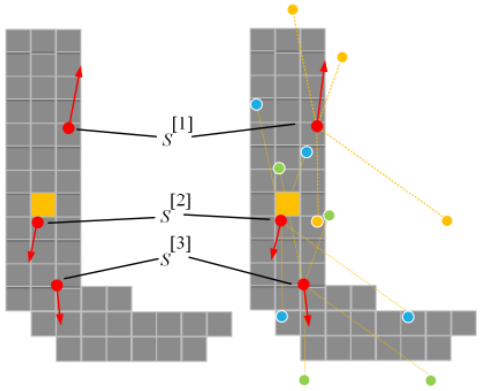

Figure 3. An intuitive example comparing two populations of particles. Both cases contain only three particles. All three particles are considered to be initialized inside the yellow grid cell and are predicted later into different other neighbor cells (here all 3 particles belong to the same tracklet created for the yellow cell). We consider that in both cases the same motion model and the same measurement model is used. However a difference is that the particles in the right image are extended with extra knowledge by incorporating 4 landmarks per particle. Intuitively one can see that in the left scenario all the particles would receive a similar weight even they were predicted into different cells. However, the right scenario gives us the possibility to better distinguish between the three predicted particles as soon as we can match the landmarks with the measured object contour. In this specific case the sample $s^{[2]}$ (with blue landmarks) should receive a higher weight due to a better allignment of the landmarks with the blob contour.

B. *Multi-channel grid estimation*

The second estimation part aims to integrate the information that is organized into grid channels. As in the current work the raw sensor data is structured within occupancy and semantic channels, at each point in time t, for each grid cell position we will define an appearance vector $\mathbf{A_t}$ described by two values: an occupancy value o_t and a semantic label l_t:

$$\mathbf{A_t} = [o_t, l_t]^T \tag{4}$$

Similarly to the previous part, the object motion and appearance estimation become joint estimation problem and can be factored as:

$$p(\mathbf{x_t}, \mathbf{A_t} | \mathbf{z_{1:t}}) = p(\mathbf{x_t} | \mathbf{z_{1:t}}) p(\mathbf{A_t} | \mathbf{x_t}, \mathbf{z_{1:t}})$$

$$= p(\mathbf{x_t} | \mathbf{z_{1:t}}) p(o_t | \mathbf{x_t}, \mathbf{z_{1:t}}) p(l_t | \mathbf{x_t}, \mathbf{z_{1:t}}) \tag{5}$$

The two terms $p(o_t | \mathbf{x_t}, \mathbf{z_{1:t}})$ and $p(l_t | \mathbf{x_t}, \mathbf{z_{1:t}})$ represent the occupancy and label posteriors that are conditioned on $\mathbf{x_t}$ – the object position and speed.

C. *Combining the two problems into one estimator*

Finally, the two update equations presented in the previous sections can be combined into one estimator as:

$$p(\mathbf{x_t}, \mathbf{A_t}, \mathbf{Q_t} | \mathbf{z_{1:t}})$$

$$= p(\mathbf{x_t} | \mathbf{z_{1:t}}) \, p(o_t | \mathbf{x_t}, \mathbf{z_{1:t}})$$

$$p(l_t | \mathbf{x_t}, \mathbf{z_{1:t}}) \prod_{k=1}^{K} p(\mathbf{q_{t,k}} | \mathbf{x_t}, \mathbf{z_{1:t}}) \tag{6}$$

Therefore, for each occupied grid cell we initialize a tracklet. The full posterior over tracklet positions, speed, landmarks and appearance components is approximated by a set of *N* particles:

$$\{\mathbf{x}_t^{[i]}, o_t^{[i]}, l_t^{[i]}, (\mathbf{q}_{t,1}^{[i]}, \Sigma_{t,1}^{[i]}) \dots, (\mathbf{q}_{t,K}^{[i]}, \Sigma_{t,K}^{[i]}), w_t^{[i]}\}_{i=1..N} \quad (7)$$

In the equation above the $\mathbf{q}_{t,k}^{[i]}$ and $\Sigma_{t,k}^{[i]}$ define the mean and 2x2 covariance of the *k*-th landmark, assigned to the *i*-th sample. We adopt an implementation based on a Rao-Blackwellised particle filter that can be summarized as follows. Each tracklet is represented by a population of particles. The particles are sampled from the position and velocity. Additionally, the appearance and landmark properties are attached to each individual sample. Both the appearance and landmark components contribute to a more precise particle weighting and are updated in closed form at the sample level.

IV. MLPT Details

There are several steps to recursively estimate the dynamic grid and its corresponding tracklets. Once new measurements are available, the following steps are applied:

1) Prediction: In the prediction step the particles are propagated according to their previous state and a linear motion model assuming a constant velocity. Additionally, each propagated sample is perturbed with a random noise component.

2) Weighting: The weighting step consists in assigning new importance weights to every predicted particle. Intuitively, a weight should reflect how likely it is that a given sample matches the observation. At a point in time *t* the measurement model is described by three individual components, a measurement cell likelihood $p\left(\mathbf{z}_t^d \middle| \mathbf{s}_t^{[i]}\right)$ computed as a distance function between the measurement cell and the nearest particle, a landmark based likelihood $p\left(\mathbf{z}_t^l \middle| \mathbf{s}_t^{[i]}\right)$ computed based on an alignment error between the measurement contours and the particle landmarks and a semantic likelihood $p\left(\mathbf{z}_t^s \middle| \mathbf{s}_t^{[i]}\right)$ given the particle's semantics. Further we will refer to these terms as *weight factors* as they will contribute to defining the overall particle importance weight. Therefore, if we consider that all three likelihood components are independent, the weight $w^{[i]}$ of the *i*-th particle $\mathbf{s}_t^{[i]}$ can be defined as:

$$w^{[i]} = p\left(\mathbf{z}_t \middle| \mathbf{s}_t^{[i]}\right) = p\left(\mathbf{z}_t^d, \mathbf{z}_t^l, \mathbf{z}_t^s \middle| \mathbf{s}_t^{[i]}\right)$$

$$= p\left(\mathbf{z}_t^d \middle| \mathbf{s}_t^{[i]}\right) p\left(\mathbf{z}_t^l \middle| \mathbf{s}_t^{[i]}\right) p\left(\mathbf{z}_t^s \middle| \mathbf{s}_t^{[i]}\right) \quad (8)$$

Next, we'll describe how the three weight components are calculated. For calculating the particle-to-measurement correspondences and distances we precompute two maps, one for defining all distances to the closest occupied points (see Fig. 4, center) and second for defining all distances to the closest measurement contours (see Fig. 4, right). Additionally, each map cell stores the position to the closest observation. The particle-to-measurement weight component is calculated as:

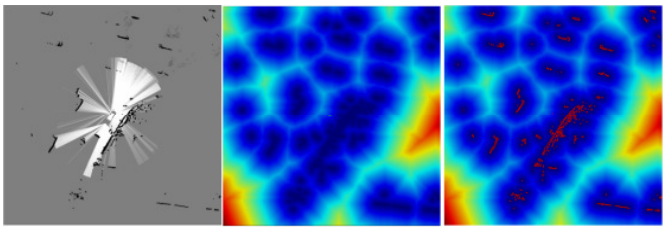

Figure 4. Occupancy grid. Center: the distance transform in which a cell has an assigned distance to the closest occupied point. The colors gradually change from blue (low distances) to red (larger distances). The occupied points (inside the object) have zero distance (dark blue). Right: a similar distance transform in which one cell stores the distance to the closest contour point. The contour points are colored with red. (the contour point color is not related to the distance transform values). It must be noted that both inside and outside object points are considered in the distance transform computation.

$$p\left(\mathbf{z}_t^d \middle| \mathbf{s}_t^{[i]}\right) = \frac{1}{\sqrt{2\pi}\sigma_d} \exp\{-\frac{d_p^2}{2\sigma_d^2}\} \quad (9)$$

where d_p is the distance between the *i*-th sample and the closest occupied point. We model an object contour point likelihood given the *k*-th particle landmark as:

$$w_l^{[k]} = \frac{1}{\sqrt{2\pi}\sigma_l} \exp\{-\frac{(d_l^{[k]})^2}{2\sigma_l^2}\} \quad (10)$$

where $d_l^{[k]}$ represents the distance between the *k*-th landmark and its corresponding contour point defined by the distance map (see Fig. 4). Thus, for a total of *K* landmarks the $p\left(\mathbf{z}_t^l \middle| \mathbf{s}_t^{[i]}\right)$ factor assigned to the *i*-th particle can be computed according to:

$$p\left(\mathbf{z}_t^l \middle| \mathbf{s}_t^{[i]}\right) = \prod_{k=1}^{K} w^{[k]} \quad (11)$$

For the semantic weight factor we first define a dissimilarity metric between the predicted label l_p (particle label) and the measurement label l_m retrieved from the closest occupied object cell:

$$d_s = 1 - \eta_l \cdot h(l_p, l_m) \quad (12)$$

Here $h(l_p, l_m)$ is a score function that is defined as:

$$h(l_p, l_m) = \begin{cases} c_1, & l_p = l_m \\ c_2, & either\ l_p\ or\ l_m\ is\ unknown \\ c_3, & l_p \neq l_m \end{cases} \quad (13)$$

and η_l is a normalization constant: $\eta_l = 1/(c_1 + c_2 + c_3)$ having c_1, c_2 and c_3 as three score values selected such that $c_3 < c_2 < c_1$. The resulted dissimilarity distance is converted into the semantic weight factor according to:

$$p\left(\mathbf{z}_t^s \middle| \mathbf{s}_t^{[i]}\right) = \frac{1}{\sqrt{2\pi}\sigma_s} \exp\{-\frac{d_s^2}{2\sigma_s^2}\} \quad (14)$$

3) Update the appearance and landmarks: As presented before, according to the Rao-Blackwellisation process, each particle has its own local landmark and appearance estimates. In order to update the particle landmarks with the newly

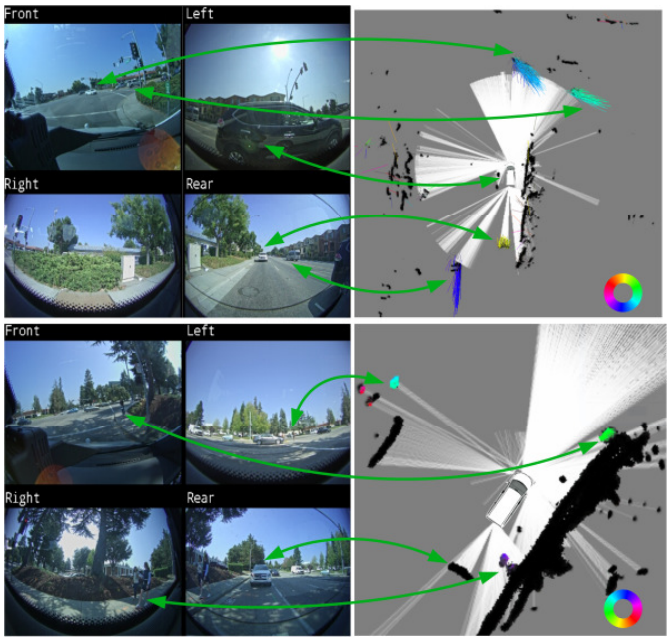

Figure 5. Left: examples from real traffic scenarios. Top-right: the estimated dynamic tracklets described by oriented speed vectors. The scene shows one stationary and four moving cars. Bottom-right: an example with the estimated grid cells (top view) depicting one stationary vehicle and several walking pedestrians. Both the dynamic tracklets and grid cells are colored based on the estimated speed orientation and magnitude. Black is for static objects, the color values are used to encode the speed orientation.

associated (closest) contour point positions we use 2x2 Kalman filters (one per landmark). It must be noted that, similarly to [20], the landmark prediction is indirectly done by the particle prediction (the landmarks are conditioned on the particle state) and follow the motion of the particle. Additionally, as also proposed in [23], for each particle its occupancy value is updated with the new measurement occupancy by using a Binary Bayes filter. However, for the simplicity, the semantic labels are kept unchanged.

4) Estimation: In the estimation step, a weighted average of the particle states is used to estimate both: the grid cell states and the tracklet states. The grid cell state is estimated by using all the particles projected into the same cell (regardless the tracklet index). However the tracklet state is estimated by considering only its own samples, even if these particles are projected into multiple cells.

5) Resampling and tracklet management: Assuming that the particle weights are normalized, for each tracklet, the resampling step selects a new set of particles from the previous set according to their importance weight by replacing the particles with lower weights. For the resampling step, a Stochastic Universal Resampling algorithm with linear complexity is used.

The last two steps of the particle filter are tracklet initialization and removal. New tracklets are initialized in measurement cells that are not sufficiently covered by particles. This is checked by computing the sum of unnormalized weights for all the particles located in the measurement cell. If the resulted sum is less than a given threshold, a new tracklet will be initialized by drawing new random hypotheses around the measurement cell.

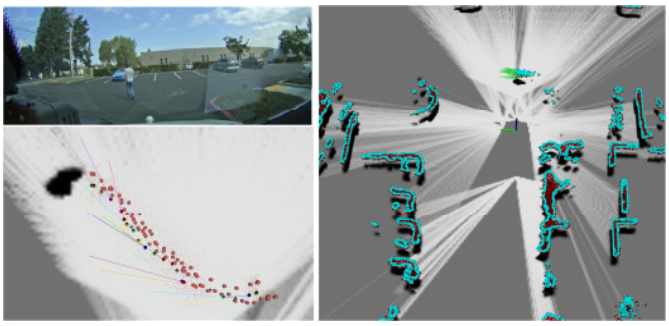

Figure 6. Top-left: a scenario with one pedestrian crossing in front of the ego-vehicle from right-near to left-far. Bottom-left: to better illustrate the estimation at the tracklet level we selected only one tracklet to be visualized and deactivated the others. The image shows the trajectory of the selected tracklet (top view). The red dots represent the estimated tracklet landmarks while the colored segments show the estimated speeds along the trajectory. Right: the visualization of the same scene (top-left image). The green vectors denote the target speeds while the cyan dots represent all the estimated landmarks from all the existing tracklets in the scene. It can be noted that these landmark positions follow the object shape.

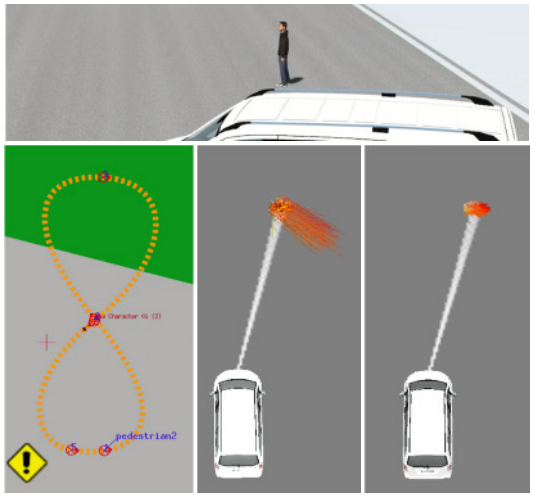

Figure 7. Top: a simulated scenario containing one pedestrian moving in an eight-shape trajectory. Bottom-left: the trajectory of the moving pedestrian. Bottom-center: the extracted dynamic tracklets represented by speed vectors. Bottom-right: the estimated dynamic grid with colored cells. Both tracklets and estimated grid cells are two alternative ways to represent the same target and are colored based on motion direction and magnitude.

The tracklets and their corresponding particles are discontinued, if they are outside the grid area, or if they are not observed or updated for a longer time.

V. Experimental Results

For the evaluation, the proposed approach was tested on various recorded traffic sequences. In addition, in order to be able to confirm the improvements in terms of the result accuracy, we have compared our method with a similar algorithm based on simulated data.

Fig. 5 presents some examples of the dynamic environment estimation for real traffic scenarios, including the estimated dynamic tracklets (top-right image) and the dynamic grid (bottom-right image). Both the dynamic tracklets and grid cells are colored based on the estimated orientation and speed. The results include various obstacles such as static cars, vehicle crossing in front of the ego-car, vehicles behind the ego-car, or pedestrians walking in the proximity.

TABLE I. SPEED AND DISTANCE ESTIMATION ACCURACY

Method	**DS-PHD [7]**		**MLPT (ours)**	
Metric	***RMSE***	***StdDev***	***RMSE***	***StdDev***
Speed	0.6884	0.2572	**0.3641**	0.3496
Distance	0.3666	0.2799	**0.3167**	0.1445
Nr. of particles	8 Mil		Approx. 8500	

Fig. 6 presents a scenario with one pedestrian crossing in front of the ego-vehicle from right to left (top image). To better illustrate the estimation at the tracklet level we activated only one tracklet to be visualized and deactivated the others. The trajectory of the selected tracklet can be seen in the bottom-left image (top view). The red dots describe the estimated tracklet landmarks, while the colored segments show the estimated speeds along the pedestrian trajectory. If we activate the visualization of all the estimated tracklets and their landmarks, then we can observe that they are explicitly describing the object shape (see Fig. 6, bottom-right image).

In order to conduct the quantitative evaluation we used a simulation environment (see Fig. 7). Basically the simulation data, replaced the input sensors and was able to provide the object position ground truth at the grid cell level. The proposed approach **M**ulti-**L**ayer **P**article Filter based **T**racking (MLPT) was compared with a similar grid-based tracking solution – the Dempster-Shafer Probability Hypothesis Density tracking for Dynamic Occupancy Grid Maps (we will refer to it as DS-PHD) [7]. The main objective of our quantitative experimental results was to analyze the algorithm convergence and their estimation accuracy in terms of root mean squared error (RMSE) and standard deviation of the estimated speed and distances. The simulated scenario included a pedestrian moving in an eight-shape trajectory (see Fig. 7) with a constant speed of 2.78 m/s (10km/h). The number of particles in the DS-PHD method was set to 8 Milion and remained fixed. However, in the current MLPT solution the number of particles was fixed to 100 particles per tracklets and depended on how many tracklets were used. Additionally, in this test we used 3 landmarks per particle. On average, the current experiment employed about 85 tracklets at a given point in time, which means 8500 particles. For the object speed calculation we selected only the cells with an estimated occupancy probability above 0.7.

The last two images in Fig. 7 (bottom-center and bottom-right) show an example of the estimated tracklets and grid cells by applying our approach. The speed estimation results of both DS-PHD and proposed MLPT approach are shown in the Fig. 8 and Table I. It can be seen that the proposed extended particle state helps the estimator to converge faster towards the ground-truth value and provides a more accurate estimation over time. It must be noted that although the DS-PHD tends to underestimate the speed in this example it provides similar values if we increase the occupancy threshold when computing the estimated speed by selecting the cells with the occupancy above 0.8. This could be explained by the fact that in the DS-PHD the occupancy is given by the particle density. The higher the particle density the higher the estimation accuracy is.

Fig. 9 presents the distance estimation. For the comparison we used the shortest distance from the ego-vehicle to the closest occupied point. Both compared methods tend to slightly underestimate the distance calculation. This is explained by the fact that particles are spread on a larger area around the target, therefore, depending on the parameter set, the estimated objects might be larger, and this is translated to a difference to up to 0.5m in distance.

Fig. 10 shows the estimated orientation. Since the orientation ground truth is not directly provided, the diagram presents comparative orientation estimations. The difference in the algorithm convergence can be also observed here. As the MLPT approach converges faster to the real object trajectory, its orientation is provided earlier (the DS-PHD plotted line is slightly delayed by being shifted to the right).

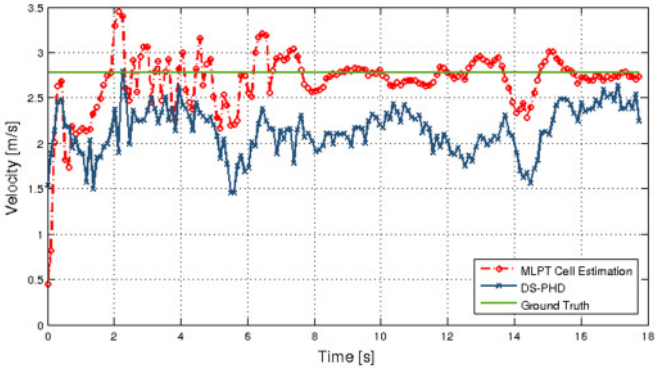

Figure 8. Top: Speed estimation. Comparison between the DS-PHD [7], an the proposed MLPT solution.

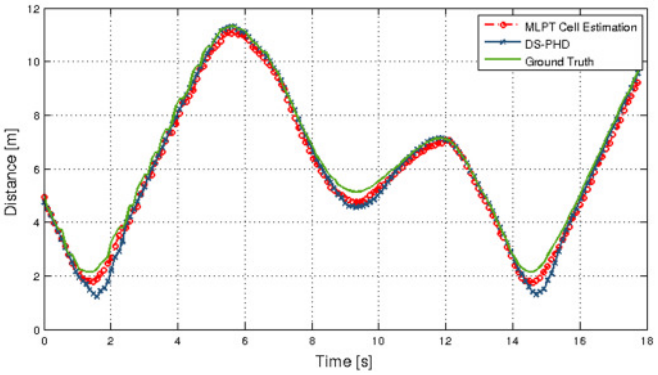

Figure 9. Top: Distance estimation. Comparison between the DS-PHD [7], an the proposed MLPT solution.

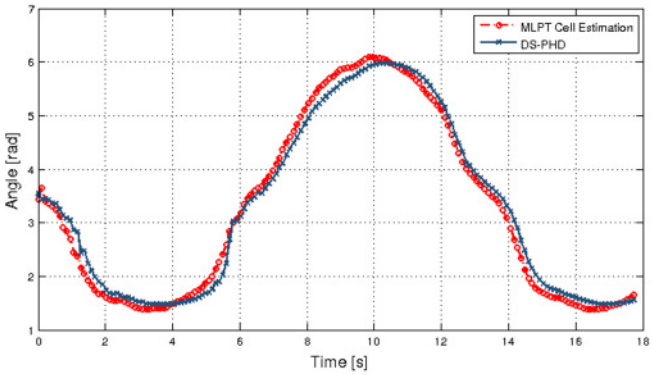

Figure 10. Top: Estimated Orientation. Comparison between the DS-PHD [7], an the proposed MLPT solution.

Table I centralizes the root mean squared error (RMSE) and standard deviation (StdDev) for the speed and distance using the two methods.

The DS-PHD algorithm is described by a highly-efficient parallel implementation (see [7] for more details), while the presented MLPT method was implemented and tested on a CPU architecture. However, based on the results presented above, one of the first observations is that the MLPT solution requires much less particles to estimate the dynamic state due to a more precise particle to measurement matching. This is explained by the fact that MLPT uses more rich particles extended with a set of landmarks which represent the partial knowledge about the shape of the tracked object hypothesis.

VI. CONCLUSIONS

This work focused on bringing new improvements in the dynamic grid map level for a better environment representation. Our main objective was to model and test a more generic and, at the same time, flexible method to track free-form environments. We developed a probabilistic solution based on a particle filter that combines two important perception tasks: fusing multi-sensor data into one estimator and stabilizing the residual errors in the position and speed estimation. The results prove that the idea of using richer particles including shape and appearance information increase the grid estimation accuracy. Although, grouping particles into individual tracklets and using hybrid filters has a potential to develop real time solutions, this topic has not been fully explored yet. One of the future works would be to focus on optimizing the current approach, as well as incorporating new grid channels such as the velocity grid computed from radar measurements.

REFERENCES

[1] A. Elfes. “Using occupancy grids for mobile robot perception and navigation”, in proc. of *Computer*, 22(6), pp.46-57, 1989.

[2] S. Sivaraman, M. M. Trivedi, "Looking at Vehicles on the Road: A Survey of Vision-Based Vehicle Detection, Tracking, and Behavior Analysis", in *IEEE Trans. on Intelligent Transportation Systems*, vol.14, no.4, pp.1773-1795, 2013.

[3] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection: A review", IEEE Trans. Pattern Anal. Mach. Intell., vol. 28, no. 5, pp. 694–711, May 2006.

[4] X. Zhang, W. Xu , C. Dong, and J.M. Dolan, “Efficient L-shape fitting for vehicle detection using laser scanners”, in Proc. of *Intelligent Vehicles Symposium (IV),* Los Angeles, CA, USA, pp. 54-59, 2017.

[5] M. H. Daraei, A. Vu and R. Manduchi, "Velocity and shape from tightly-coupled LiDAR and camera," *in Proc. of 2017 IEEE Intelligent Vehicles Symposium (IV)*, Los Angeles, CA, USA, pp. 60-67, 2017.

[6] A. Petrovskaya and S. Thrun, “Model based vehicle detection and tracking for autonomous urban driving”, in *Autonomous Robots*, vol. 26, no. 2-3, pp. 123-139, 2009.

[7] D. Nuss, et al., "A random finite set approach for dynamic occupancy grid maps with real-time application”, arXiv preprint arXiv:1605.02406

[8] D. Nuss, T. Yuan, G. Krehl, M. Stuebler, S. Reuter and K. Dietmayer, "Fusion of laser and radar sensor data with a sequential Monte Carlo Bayesian occupancy filter," 2015 IEEE Intelligent Vehicles Symposium (IV), Seoul, 2015, pp. 1074-1081.

[9] F. Kunz et al., "Autonomous driving at Ulm University: A modular, robust, and sensor-independent fusion approach," 2015 IEEE Intelligent Vehicles Symposium (IV), Seoul, 2015, pp. 666-673.

[10] R. Varga, A. Costea, H. Florea, I. Giosan and S. Nedevschi, "Super-sensor for 360-degree environment perception: Point cloud segmentation using image features," 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, 2017, pp. 1-8.

[11] R. Danescu, F. Oniga, S. Nedevschi, “Modeling and Tracking the Driving Environment with a Particle Based Occupancy Grid”, IEEE Transactions on Intelligent Transportation Systems, vol. 12, No. 4, pp. 1331-1342, 2011.

[12] R. Danescu, S. Nedevschi, “A Particle-Based Solution for Modeling and Tracking Dynamic Digital Elevation Maps”, IEEE Transactions on Intelligent Transportation Systems, vol. 15, No. 3, pp. 1002-1015, June 2014.

[13] A. Nègre, L. Rummelhard and C. Laugier, "Hybrid sampling Bayesian Occupancy Filter," 2014 IEEE Intelligent Vehicles Symposium Proceedings, Dearborn, MI, 2014, pp. 1307-1312.

[14] S. Steyer, G. Tanzmeister and D. Wollherr, "Object tracking based on evidential dynamic occupancy grids in urban environments," 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, 2017, pp. 1064-1070.

[15] A. Geiger, P. Lenz and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, 2012, pp. 3354-3361.

[16] L. Schneider et al., "Semantic Stixels: Depth is not enough," 2016 IEEE Intelligent Vehicles Symposium (IV), Gothenburg, 2016, pp. 110-117

[17] D. Pfeiffer and U. Franke, "Efficient Representation of Traffic Scenes by Means of Dynamic Stixels," 2010 IEEE Intelligent Vehicles Symposium, San Diego, CA, 2010, pp. 217-224.

[18] Jackson, J.D.; Yezzi, A.J.; Soatto, S., "Tracking deformable moving objects under severe occlusions," Decision and Control, CDC. 43rd IEEE Conference on , vol.3, no., pp.2990,2995 Vol.3, 14-17 Dec. 2004

[19] M. Isard and A. Blake “Condensation – conditional density propagation for visual tracking” in International Journal of Computer Vision, 29(1):5–28, 1998.

[20] A. Vatavu, R. Danescu, and S. Nedevschi, "Stereovision-Based Multiple Object Tracking in Traffic Scenarios using Free-Form Obstacle Delimiters and Particle Filters", IEEE Transactions on Intelligent Transportation Systems, Vol. 16, No. 1, pp. 498-511, 2015.

[21] U. Franke, C. Rabe, H. Badino, and S. Gehrig, “6d-vision: Fusion of stereo and motion for robust environment perception,” in 27th Annual Meeting of the German Association for Pattern Recognition DAGM ’05, 2005, pp. 216-223.

[22] A. Broggi, S. Cattani, M. Patander, M. Sabbatelli and P. Zani, "A full 3D voxel-based dynamic obstacle detection for urban scenario using stereo vision", in Proc. of IEEE ITSC 2013, pp.71-76, 6-9 Oct. 2013.

[23] A. Vatavu, R. Danescu, and S. Nedevschi, "Modeling and Tracking of Crowded Traffic Scenes by using Policy Trees, Occupancy Grid Blocks and Bayesian Filters", in Proceedings of 17th International IEEE Conference Intelligent Transportation Systems (ITSC 2014), Chingdao, China, October 9-11, 2014.

[24] S. Kraemer, M. E. Bouzouraa and C. Stiller, "Simultaneous tracking and shape estimation using a multi-layer laserscanner," in IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, 2017, pp. 1-7.

[25] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, “FastSLAM: A factored solution to the simultaneous localization and mapping problem”, in Aaai/iaai, 593598, 2002

[26] C. Coue, C. Pradalier, C. Laugier, T. Fraichard, and P. Bessiere, “Bayesian occupancy filtering for multitarget tracking: an automotive application”, in The International Journal of Robotics Research, vol. 25, no. 1, pp. 2006, 19–30.

[27] A. Doucet, N. de Freitas, K. Murphy, and S. Russell. Rao-blackwellised particle filters for dynamic bayes nets. In Proceedings of Conference on Uncertainty in Artificial Intelligence, 2000

[28] C. Wöhler, O. Schumann, M. Hahn and J. Dickmann, "Comparison of random forest and long short-term memory network performances in classification tasks using radar," 2017 Sensor Data Fusion: Trends, Solutions, Applications (SDF), Bonn, 2017, pp. 1-6.