2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Grid Mapping in Dynamic Road Environments: Classification of Dynamic Cell Hypothesis via Tracking

Matthias Schreier, Volker Willert, Jürgen Adamy

Abstract— **We propose a method capable of acquiring an occupancy grid map-based representation of the local, static driving environment around an intelligent vehicle in the presence of dynamic objects. These corrupt the representation due to violating the underlying static-world assumptions of common grid mapping algorithms and are therefore detected and filtered from the map. For this purpose, a subsequent step is suggested that identifies, clusters and merges dynamic cell hypothesis in a novel way. Thereafter, an Interacting-Multiple-Model-Unscented-Kalman-Probabilistic-Data-Association (IMM-UK-PDA) tracker is used to classify of whether cell movements behave consistently with possible movement characteristics of real dynamic objects or are just generated by noise or newly observed static environment. In opposition to many other approaches, the method explicitly combines information of newly occupied and free areas, completes the shape of only partly visible dynamic objects and uses an advanced object tracking scheme to clean the grid from dynamic object corruptions. The method is evaluated with grids generated by an automotive radar and stereo camera in real traffic environments.**

I. INTRODUCTION AND RELATED WORK

Nowadays, more and more occupancy grid maps are used as a dense environment representation for Advanced Driver Assistance Systems (ADAS) [1], [2] and intelligent vehicles in order to fulfill tasks such as automated lateral vehicle guidance with respect to elevated objects or collision avoidance under consideration of drivable free space [3], [4]. These are, however, normally based on mapping algorithms designed for a static world, e.g. Binary Bayes Filters, so that dynamic objects corrupt the representation in form of trails of slowly declining occupancy probabilities.[1] Although, in principle, occupancy grids can deal with dynamic environments, they have an inherent drawback: To unlearn that a cell is free takes as many observations as to learn its occupancy [6]. The aim of the paper is to present a method that minimizes these errors online to generate a local, static grid map valid at the moment it is generated in general driving environments. This facilitates higher-level environment recognition tasks that do not have to deal with moving object corruptions any more as well as extend available static free space for trajectory planners.

Previous works that tackle the problem of mapping highly dynamic environments can be subdivided into two categories. First, algorithms that try to incorporate the scene dynamics directly in the mapping process and second, methods that filter out measurements supposed to belong to dynamic objects and subsequently use a standard mapping algorithm suitable for a static world.

A representative of the first category is the so-called Bayesian Occupancy Filter [7] that generates a four-dimensional grid representation of the environment – two dimensions for the Cartesian coordinates and two for the velocity components of each cell – in which each cell movement is predicted with a constant velocity model. A computationally less demanding modification [8] produces a two-dimensional occupancy grid representation with each cell having an associated distribution over possible velocities. Another approach is described in [9] where a particle filter-based tracking is used to infer the velocity of each grid cell. The raw occupancy grid serves as the measurement vector and the particles additionally provide the building blocks of the modeled world. Two grids are used in [10], one for representing the occupancy probability of moving and one of static objects, which are mapped with different inverse sensor models. However, no physically interpretable prediction models are applied. Hence, if a moving object cannot be detected, the area is inevitably mapped into the static grid, because detection fails cannot be bridged.

Apart from the necessary processing power, the disadvantage of these methods is that they cannot easily be combined with existing systems that rely on a standard grid representation, contrary to the works of the second category.

Representatives of this second group are the works of [11] that bring up the general problem of SLAMMOT (Simultaneous Localization, Mapping and Moving Object Tracking) and provide a computationally feasible solution by decomposing the problem into two separate estimators that solve SLAM (Simultaneous Localization And Mapping) and DATMO (Detection And Tracking of Moving Objects) concurrently, which results in separate moving object grids for each object as well as a grid for the static world. Either a consistency-based or a moving-object map-based detector is used to classify measurements belonging to moving objects directly. These are filtered from the static map to get a better pose estimation within SLAM. In [12], moving people are tracked by means of a sample-based JPDA filter and measurement beams resulting from moving objects are likewise excluded to improve the map quality. In [13], new measurements are compared with the local map to classify them as dynamic. However, ambiguities occur if objects appear in unexplored regions, because it is not possible to distinguish between moving and newly appearing static environments. There-

M. Schreier, V. Willert and J. Adamy are with the Institute of Automatic Control and Mechatronics, Control Theory and Robotics Lab, TU Darmstadt, Landgraf-Georg-Str. 4, 64283 Darmstadt, Germany {schreier,vwillert,jadamy}@rtr.tu-darmstadt.de

[1] A similar problem arises in computer vision algorithms that filter the optical flow [5].

fore, a tracking step is used in [14] as an improvement after initial dynamic cell hypothesis generation to separate dynamic and static objects more robustly. Laser scanner measurements are subsequently connected with the help of box models of predefined sizes for different classes such as pedestrians or vehicles. A motion evidence score is calculated in [15] to help filtering out false moving object hypothesis. This score is based on newly occupied and free regions, but, in opposition to our approach, the information is not used within the measurement vector generation for an employed Rao-Blackwellized particle filter-based tracking. A combination of object-based multi-target tracking and grid mapping can further be found in [16], where grid cells are associated to object states and predicted within the map. A subsequent feedback from the map is used to preserve consistency between the map content and the object list. With this laser scanner-based system, trails of moving objects can be suppressed and the tracking itself benefits from the grid mapping. In [17], the disparity image is classified into regions of stationary and moving parts by generating object hypotheses by segmenting object contours on a v-disparity image that are tracked over time with additionally exploiting optical flow within the measurement equations. A classifier based on a sequential probability ratio test on the residuals of a stationary and a moving process model is used to distinguish moving from stationary parts of the environment, whereupon only non-moving parts of the disparity image are integrated into the grid. In [18], fuzzy rules are used for the distinction between static and dynamic cells that take into account criteria such as cluster size, correlation, grid resolution, etc. Once a cell is classified as dynamic, it is, however, not updated in the standard way, but by taking the current sensor measurement as the new occupancy probability. Due to unconvincing results, the same authors propose a different approach within [19]. The grid itself undergoes a hierarchical segmentation based on a distance criterion, whereupon each segment is tried to be associated with already tracked objects. Segments associated with moving objects are used to update an IMM tracking filter while only the ones associated with stationary objects update an occupancy grid. The problem of direct grid segmentation, as done here, is that adjacent static and moving segments merge together, which limits the approach in unstructured inner city environments with many static segments in the vicinity of moving objects.

All in all, the main limitation of the methods of this second category is that they can only be applied if measurements can robustly be separated into measurements of stationary and moving objects.[2] This is a challenging task even for high-precision laser scanner sensors such as utilized in [11], [14], [15], yet alone for less accurate distance sensors like stereo cameras and radars (we use in our work) as already stated in [19].

[2]Some vision-based works, e.g. [20], circumvent this separation problem by using trained image feature-based vehicle/people detectors and exclude all emerging tracks from a simultaneously built static occupancy grid. This is, however, neither possible with other environment sensors nor for non-trained dynamic objects and would also exclude standing vehicles/people.

Our approach belongs to the second category, therefore keeps the standard grid mapping algorithm, but differs from the mentioned methods in a couple of ways. First, we do not try to classify raw sensor measurements as static or dynamic, but cells within temporally consecutive maps after a traditional grid map update step. Therefore, false, spurious measurements are filtered out by the inverse sensor model initially. Second, we explicitly make use of a measurement of newly available free space and combine this information with newly occupied areas for general dynamic object hypothesis generation even in the vicinity of static environment. With this novel approach, it is not only possible to receive a unique object angle measurement within each time step, but also to recombine even only partly visible objects without the need of clustering the original grid or measurement. Third, a nonlinear, adaptive Interacting-Multiple-Model-Unscented-Kalman-Probabilistic-Data-Association (IMM-UK-PDA) tracker is used not merely to recursively estimate the states of underlying motion models of extended objects, but rather for checking the consistency of cell movements with movement characteristics of real dynamic objects. This helps in deciding between real dynamics, newly mapped static environment and sensor noise. Its further possible to filter out dynamic cells even if the motion detector fails by using the prediction models.

II. Theoretical Foundations – IMM-UK-PDA

This section provides the theoretical foundations of a combination of an Interacting Multiple Model (IMM) multi-target tracking filter with underlying Unscented Kalman (UK) filters and additional Probabilistic Data Association (PDA) used for the classification. It is mainly based upon a combination of [21]–[25] and provides a computationally modest solution to the problem of recursively estimating states and mode probabilities of targets, each described by a jump Markov nonlinear system, in the presence of clutter.[3]

We consider the j-th model of a model set $\mathcal{M} = \{M_j\}_{j=1}^{r}$ to be given by the nonlinear stochastic state space model

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}_j(\boldsymbol{x}_k, \boldsymbol{u}_k) + \boldsymbol{w}_{j,k}, \tag{1a}$$

$$\boldsymbol{z}_k = \boldsymbol{h}_j(\boldsymbol{x}_k, \boldsymbol{u}_k) + \boldsymbol{v}_{j,k}, \tag{1b}$$

with input vector $\boldsymbol{u}_k \in \mathbb{R}^p$, state vector $\boldsymbol{x}_k \in \mathbb{R}^n$, measurement vector $\boldsymbol{z}_k \in \mathbb{R}^q$, system function $\boldsymbol{f}_j$ and measurement function $\boldsymbol{h}_j$ at each time step k. The zero-mean, white, Gaussian noise sequences $\boldsymbol{w}_{j,k} \in \mathbb{R}^n$ and $\boldsymbol{v}_{j,k} \in \mathbb{R}^q$ are mutually independent with covariance matrices $\boldsymbol{Q}_{j,k}$ and $\boldsymbol{R}_{j,k}$, respectively. The evolution of the hybrid system among the r models is considered a first order Markov chain with time-invariant Markovian model transition probability matrix

$$\boldsymbol{\Pi} = \begin{pmatrix} p_{11} & \cdots & p_{r1} \\ \vdots & \ddots & \vdots \\ p_{1r} & \cdots & p_{rr} \end{pmatrix} \in \mathbb{R}^{r \times r}, \tag{2}$$

[3]The term mode probability is used for the probability of the event that a hypothesized model matches the true behavior pattern (or mode) of a target.

in which p_{ij} denotes the probability that a mode transition occurs from model i to model j [21]. The explanation of the tracking filter is subdivided into five main steps – the interaction (or mixing) step, the state prediction and measurement validation step, the data association and model-specific filtering step, the mode probability update step and the combination step.

A. Interaction Step

In the interaction step, the individual state and covariance estimates of the underlying filters of the last time step $\hat{\boldsymbol{x}}_{j,k-1}, \boldsymbol{P}_{j,k-1}$ are probabilistically mixed to form a single initial state $\hat{\boldsymbol{x}}^{\star}_{j,k-1}$ and covariance $\boldsymbol{P}^{\star}_{j,k-1}$ for each filter j according to

$$\hat{\boldsymbol{x}}^{\star}_{j,k-1} = \sum_{i=1}^{r} \mu_{(i|j),k-1} \hat{\boldsymbol{x}}_{i,k-1}, \tag{3a}$$

$$\begin{aligned}\boldsymbol{P}^{\star}_{j,k-1} = \sum_{i=1}^{r} \mu_{(i|j),k-1} [\boldsymbol{P}_{i,k-1} + \\ + (\hat{\boldsymbol{x}}_{i,k-1} - \hat{\boldsymbol{x}}^{\star}_{j,k-1})(\hat{\boldsymbol{x}}_{i,k-1} - \hat{\boldsymbol{x}}^{\star}_{j,k-1})^T].\end{aligned} \tag{3b}$$

Herein, $\mu_{(i|j),k-1}$ are the conditional mode probabilities or mixing probabilities, corresponding to the probability that mode i has been in effect in the previous cycle given that mode j is active in the current time step. They are calculated according to

$$\mu_{(i|j),k-1} = \frac{p_{ij}\mu_{i,k-1}}{\mu^{-}_{j,k}} \tag{4}$$

with the help of the a priori mode probabilities $\boldsymbol{\mu}^{-}_{k} = (\mu^{-}_{1,k}, \dots, \mu^{-}_{r,k})^T$ of the actual time step, which are themselves given by the prediction of the mode probabilities of the last time step with elements from $\boldsymbol{\Pi}$ as follows:

$$\mu^{-}_{j,k} = \sum_{i=1}^{r} p_{ij}\mu_{i,k-1}. \tag{5}$$

Consequently, the initial state vector of each filter results from the state estimation of all filters of the last time step.

B. Prediction and Measurement Validation Step

Afterwards, these initial states and covariances are predicted to the actual time step k for each model individually in order to get predicted measurements. Due to nonlinearities, this is realized by means of an UK filter prediction [25]:[4]

$$\begin{aligned}\boldsymbol{\mathcal{X}}_{k-1} &= \left(\hat{\boldsymbol{x}}^{\star}_{k-1} \quad \gamma\sqrt{\boldsymbol{P}^{\star}_{k-1}} \oplus \hat{\boldsymbol{x}}^{\star}_{k-1} \quad -\gamma\sqrt{\boldsymbol{P}^{\star}_{k-1}} \oplus \hat{\boldsymbol{x}}^{\star}_{k-1}\right), \\ \boldsymbol{\mathcal{X}}^{*}_{i,k} &= \boldsymbol{f}(\boldsymbol{\mathcal{X}}_{i,k-1}, \boldsymbol{u}_{k-1}), \qquad i = 0, \dots, 2n, \\ \hat{\boldsymbol{x}}^{-}_{k} &= \sum_{i=0}^{2n} w^{(m)}_{i} \boldsymbol{\mathcal{X}}^{*}_{i,k}, \\ \boldsymbol{P}^{-}_{k} &= \sum_{i=0}^{2n} w^{(c)}_{i} \left(\boldsymbol{\mathcal{X}}^{*}_{i,k} - \hat{\boldsymbol{x}}^{-}_{k}\right)\left(\boldsymbol{\mathcal{X}}^{*}_{i,k} - \hat{\boldsymbol{x}}^{-}_{k}\right)^T + \boldsymbol{Q}_{k-1},\end{aligned}$$

$$\begin{aligned}\boldsymbol{\mathcal{X}}_{k} &= \left(\hat{\boldsymbol{x}}^{-}_{k} \quad \gamma\sqrt{\boldsymbol{P}^{-}_{k}} \oplus \hat{\boldsymbol{x}}^{-}_{k} \quad -\gamma\sqrt{\boldsymbol{P}^{-}_{k}} \oplus \hat{\boldsymbol{x}}^{-}_{k}\right), \\ \boldsymbol{\mathcal{Z}}_{i,k} &= \boldsymbol{h}(\boldsymbol{\mathcal{X}}_{i,k}, \boldsymbol{u}_{k}), \qquad i = 0, \dots, 2n, \\ \hat{\boldsymbol{z}}^{-}_{k} &= \sum_{i=0}^{2n} w^{(m)}_{i} \boldsymbol{\mathcal{Z}}_{i,k}, \\ \boldsymbol{S}_{k} &= \sum_{i=0}^{2n} w^{(c)}_{i} \left(\boldsymbol{\mathcal{Z}}_{i,k} - \hat{\boldsymbol{z}}^{-}_{k}\right)\left(\boldsymbol{\mathcal{Z}}_{i,k} - \hat{\boldsymbol{z}}^{-}_{k}\right)^T + \boldsymbol{R}_{k}.\end{aligned} \tag{6}$$

At first, $2n+1$ sigma points $\boldsymbol{\mathcal{X}}_i$ with $i \in \{0, \dots, 2n\}$, subsumed in the matrix $\boldsymbol{\mathcal{X}} = (\boldsymbol{\mathcal{X}}_0, \dots, \boldsymbol{\mathcal{X}}_{2n}) \in \mathbb{R}^{n \times (2n+1)}$, are chosen based on a square-root decomposition of the mixed initial covariance of each filter. Each sigma point has two associated scalar weights $w^{(m)}_i$ and $w^{(c)}_i$ given by

$$w^{(m)}_0 = \frac{\lambda}{n+\lambda}, \quad w^{(c)}_0 = \frac{\lambda}{n+\lambda} + (1 - \alpha^2_U + \beta_U), \tag{7a}$$

$$w^{(m)}_i = w^{(c)}_i = \frac{1}{2(n+\lambda)}, \quad i = 1, \dots, 2n, \tag{7b}$$

$$\lambda = \alpha^2_U(n + \kappa_U) - n, \tag{7c}$$

$$\gamma = \sqrt{n+\lambda}, \tag{7d}$$

with scaling parameters $\alpha_U, \beta_U, \kappa_U, \lambda$ and γ. These sigma points are propagated through the system function $\boldsymbol{f}$ without approximation and the mean and variance are retrieved with the calculated weights. With $\boldsymbol{Q}_{k-1}$, new sigma points are chosen, propagated through the measurement function $\boldsymbol{h}$ and the predicted, a priori measurement $\hat{\boldsymbol{z}}^{-}_{k}$ with associated innovation covariance matrix $\boldsymbol{S}_k$ is finally recovered from the propagated sigma points $\boldsymbol{\mathcal{Z}}_i$ for each model.

For validating a measurement for all models of a track, it must lie within an elliptical validation gate

$$\mathcal{V}_k(\gamma_G) = \{\boldsymbol{z}_k \mid (\boldsymbol{z}_k - \hat{\boldsymbol{z}}^{-}_{j_r,k})^T \boldsymbol{S}^{-1}_{j_r,k} (\boldsymbol{z}_k - \hat{\boldsymbol{z}}^{-}_{j_r,k}) \leq \gamma_G\}, \tag{8}$$

centered at $\hat{\boldsymbol{z}}^{-}_{j_r,k}$, with gate threshold γ_G [21] and j_r normally calculated as $j_r = \arg\max_{j \in \mathcal{M}} |\boldsymbol{S}_{j,k}|$ [23]. The threshold γ_G is obtained from the inverse chi-square cumulative distribution with a chosen gate probability $P_G = P(\boldsymbol{z}_k \in \mathcal{V}_k(\gamma_G))$ and $\dim(\boldsymbol{z}) = q$ degrees of freedom. The volume of the validation region $\mathcal{V}_k$ is $V_k = c_q |\gamma_G \boldsymbol{S}_{j_r,k}|^{\frac{1}{2}}$ with the volume of the q-dimensional unit hypersphere $c_q = \frac{\pi^{\frac{1}{2}q}}{\Gamma(\frac{1}{2}q+1)}$ and gamma function Γ [21]. Consequently, we get one validated measurement set $\mathcal{Z}_{v,k} = \{\boldsymbol{z}_{m,k}\}^{N_v}_{m=1}$ for all models of a track.

C. Data Association and Model-Specific Filtering Step

This is followed by a model-specific filtering by updating each track with the r UK-PDA filters that yield a posteriori state and covariance estimates $\hat{\boldsymbol{x}}_{j,k}$ and $\boldsymbol{P}_{j,k}$ for each model by incorporating the actual validated measurement set:[5]

$$\begin{aligned}\boldsymbol{C}_{x_k,z_k} &= \sum_{i=0}^{2n} w^{(c)}_i \left(\boldsymbol{\mathcal{X}}_{i,k} - \hat{\boldsymbol{x}}^{-}_{k}\right)\left(\boldsymbol{\mathcal{Z}}_{i,k} - \hat{\boldsymbol{z}}^{-}_{k}\right)^T, \\ \boldsymbol{K}_k &= \boldsymbol{C}_{x_k,z_k} \boldsymbol{S}^{-1}_{k},\end{aligned}$$

[4]For shorter notation, the operation of adding a column vector $\boldsymbol{b}$ to each column of a matrix $\boldsymbol{A}$ is written as $\boldsymbol{A} \oplus \boldsymbol{b}$. Moreover, the index for each model j is omitted.

[5]The index for each model j is omitted here for brevity.

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k \boldsymbol{v}_k = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k \underbrace{\sum_{m=1}^{N_v} \beta_{m,k} \underbrace{(\boldsymbol{z}_{m,k} - \hat{\boldsymbol{z}}_k^-)}_{\boldsymbol{v}_{m,k}}}_{\boldsymbol{v}_k},$$

$$\boldsymbol{P}_k = \beta_{0,k}\boldsymbol{P}_k^- + (1-\beta_{0,k})(\boldsymbol{P}_k^- - \boldsymbol{K}_k\boldsymbol{S}_k\boldsymbol{K}_k^T) + \\ + \boldsymbol{K}_k \left(\sum_{m=1}^{N_v} \beta_{m,k}\boldsymbol{v}_{m,k}\boldsymbol{v}_{m,k}^T - \boldsymbol{v}_k\boldsymbol{v}_k^T \right) \boldsymbol{K}_k^T. \tag{9}$$

First, the cross covariance matrix $\boldsymbol{C}_{x_k,z_k}$ between predicted measurements and states are used together with innovation covariance $\boldsymbol{S}_k$ to calculate the optimal Kalman gain $\boldsymbol{K}_k$ for each model such as within the normal UK filter. The state and covariance update step then equals the conventional PDA filter update [21] that weights the influence of a measurement $\boldsymbol{z}_{m,k}$ with associated innovation $\boldsymbol{v}_{m,k}$, according to its association probability $\beta_{m,k}$. We consider a nonparametric, uniform, diffuse prior clutter model that uses the number of returns in the track gate to estimate the clutter density and that is suitable for heterogeneous clutter environments as any number of false measurements is considered equiprobable [23]. Under these considerations, the association probability $\beta_{m,k}$ for $\boldsymbol{z}_{m,k}$ being the correct measurement given a target detection probability P_D is calculated according to

$$\beta_{m,k} = \begin{cases} \frac{e_m}{b+\sum_{i=1}^{N_v} e_i}, & m = 1,\dots,N_v, \\ \frac{b}{b+\sum_{i=1}^{N_v} e_i}, & m = 0, \end{cases} \\ e_m = e^{-\frac{1}{2}\boldsymbol{v}_{m,k}^T \boldsymbol{S}_k^{-1} \boldsymbol{v}_{m,k}}, \\ b = \left(\frac{2\pi}{\gamma}\right)^{\frac{q}{2}} \frac{N_v(1-P_D P_G)}{c_q P_D}, \tag{10}$$

while $\beta_{0,k}$ is the probability that no measurements within the gate are correctly belonging to the track and model under consideration [21].

D. Mode Probability Update Step

With the Gaussian-uniform mixture model likelihoods

$$\lambda_{j,k} = \frac{1-(P_D P_G)}{(V_k)^{N_v}} + \frac{P_D V_k^{(1-N_v)}}{N_v \sqrt{|2\pi \boldsymbol{S}_{j,k}|}} \sum_{m=1}^{N_v} e_{j,m}, \tag{11}$$

the mode probabilities can now be updated based on how well the measurements fit to the model as follows [21]:

$$\mu_{j,k} = \frac{\bar{\mu_{j,k}}\lambda_{j,k}}{\sum_{i=1}^{r} \bar{\mu_{i,k}}\lambda_{i,k}}. \tag{12}$$

E. Combination Step

Just for final output purpose, the individual updated filter states are combined to a final state and covariance estimate

$$\hat{\boldsymbol{x}}_k = \sum_{j=1}^{r} \mu_{j,k}\hat{\boldsymbol{x}}_{j,k}, \tag{13a}$$

$$\boldsymbol{P}_k = \sum_{j=1}^{r} \mu_{j,k}[\boldsymbol{P}_{j,k} + (\hat{\boldsymbol{x}}_{j,k} - \hat{\boldsymbol{x}}_k)(\hat{\boldsymbol{x}}_{j,k} - \hat{\boldsymbol{x}}_k)^T]. \tag{13b}$$

In the next section, it will be made clear, why this filter combination is suited to robustly classify the dynamic status of grid cells online.

III. Algorithm

A complete overview of the proposed algorithm is given in Fig. 1. It consists of three main steps: The generation

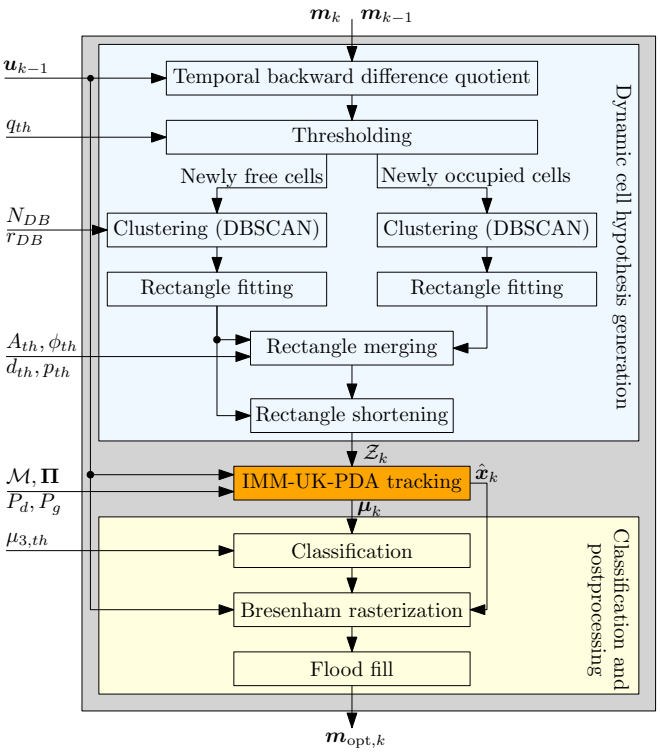

Fig. 1. Overview of one time step of the proposed algorithm.

of dynamic cell hypothesis (blue), the subsequent tracking (orange) as well as the classification and grid post processing (yellow), which are described in detail in the following subsections and further illustrated in the running example in Fig. 2. The actual grid map $\boldsymbol{m}_k$, the map of the last time step $\boldsymbol{m}_{k-1}$ as well as the ego motion vector $\boldsymbol{u}_{k-1}$ are the inputs of the algorithm that outputs an optimized grid $\boldsymbol{m}_{\text{opt},k}$, free of dynamic object corruptions. The grid maps themselves are supposed to be the result of standard individual Binary Bayes Filter updates [26] for each cell m_i. The set of environment sensor measurements is generated by a stereo camera and an automotive radar sensor with individual appropriate inverse sensor models in our case, the set of ego vehicle poses is known precisely due to the ego vehicles' ESP sensors. As this common kind of mapping assumes a static world, dynamic objects show trails of slowly declining occupancy probabilities (Fig. 2(a,b)).

A. Dynamic Cell Hypothesis Generation

The first step within the grid-based dynamic cell hypothesis generation is the calculation of the ego-motion compensated temporal backward difference quotient (Fig. 2c) of the occupied probability for each grid cell to highlight changing

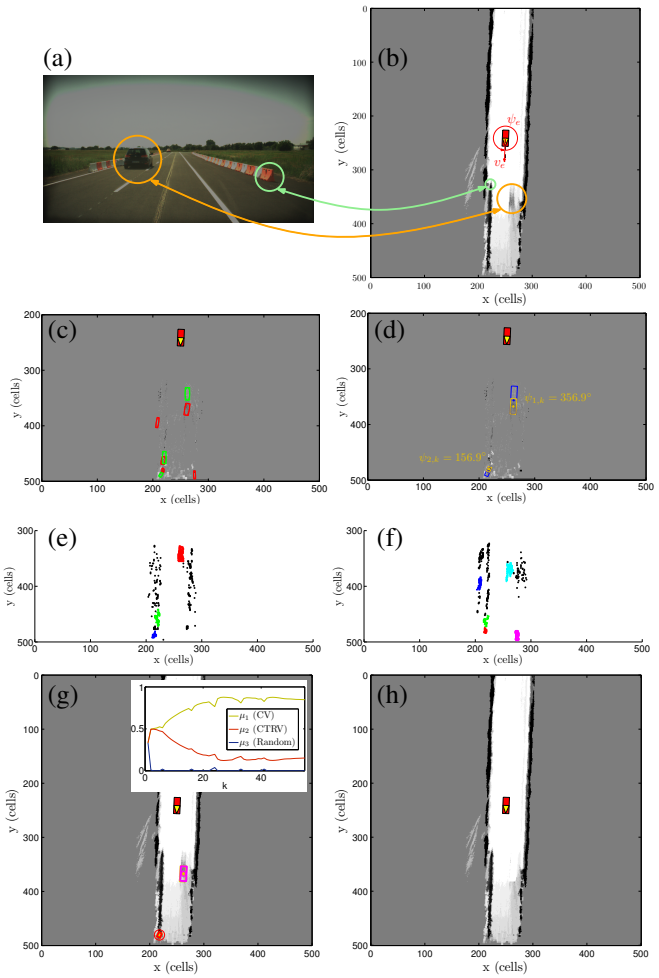

Fig. 2. The exemplary driving scene (a) contains a dynamic object in front of the ego vehicle and construction walls at the side. The unoptimized occupancy grid map $\boldsymbol{m}_k$ (b) is acquired from an automotive radar and a stereo camera. The darker a cell, the higher its occupancy probability $p(m_i)$. The map is of fixed size $(80\,\mathrm{m} \times 80\,\mathrm{m})$ and shifted with the ego vehicle's motion, whose orientation in the map is given by the yaw angle ψ_e. The temporal difference quotient with corresponding individual rectangles (newly free: green; newly occupied: red) is visualized in (c), the combined rectangle (blue) and reconstructed dynamic object hypothesis (orange) in (d), clustering results of newly free/occupied areas in (e/f) and tracks in (g). The random mode probability of the real dynamic object rapidly decreases towards zero, which is therefore classified as dynamic in opposition to the false hypothesis at lower left. The final, optimized map $\boldsymbol{m}_{\mathrm{opt},k}$ (h) only contains the static world.

regions, which could stem from real dynamic objects, but also from newly mapped static environment or noise. Frame differencing is a common approach for movement detection in the image analysis domain, the problem, however, is that the main parts of moving objects are suppressed and invisible in this representation. This is due to the fact that the occupancy probability changes only marginally in areas, where the dynamic objects really are, but much stronger in areas that the objects just entered and just passed. Consequently, dynamic objects have to be reconstructed. Therefore, two new binary grids are generated by segmenting the temporal difference grid with two thresholds $\pm q_{th}$, so that one contains only cells of rising free probability (newly free cells) and the other only cells of rising occupancy probability (newly occupied cells). These new grids are clustered separately (Fig. 2(e,f)) by means of a Density Based Spatial Clustering for Applications with Noise (DBSCAN) with neighborhood radius r_{DB} and minimal point number N_{DB} [27], followed by a calculation of oriented rectangles for each individual cluster (Fig. 2c). This is made more robust by first extracting the error ellipse of each cluster by scaling its covariance matrix and generating the bounding rectangle for each ellipse. Afterwards, rectangles of newly free clusters are tried to be merged with rectangles of newly occupied clusters in order to bridge invisible object areas. This rectangle shape model assumption is reasonable for vehicles, bicycles or pedestrians and makes a completion of only partially visible objects possible, e.g. if only two edges are visible. The merging criteria are deduced from considerations that hold in perfectly mapped bird's eye view grids. First, newly occupied and newly free rectangles belonging to the same object must have the same area as dynamic objects are supposed to be non-deforming. Second, their orientation must be equal. Third, cells that lie in between have to be occupied in the original grid and fourth, their distance must lie within certain bounds corresponding to real object sizes. Based on a relaxed fulfillment of these criteria, rectangles are merged by calculating the combined rectangle of newly free and newly occupied areas of an object. These are, however, too large as the areas of newly free cells do not correspond to the real object and therefore have to be shortened by the newly free rectangle in order to generate a final dynamic object hypothesis (Fig. 2d). The steps are further elaborated in Fig. 3. For each final object rectangle, the center x, y, the

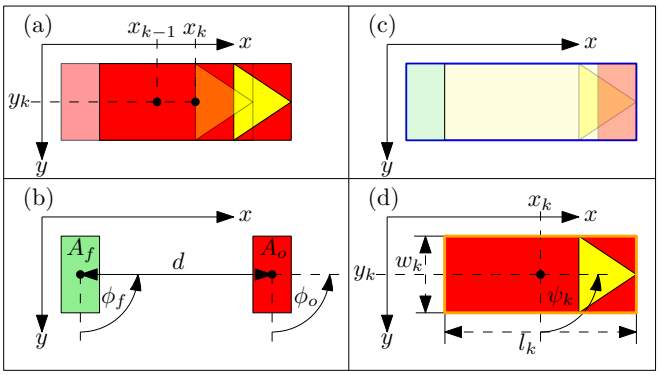

Fig. 3. The position of a dynamic object moving to the right (yellow arrow) is visualized in (a) within two consecutive time steps in a grid map fixed coordinate system. The corresponding temporal backward difference quotient grid is shown in (b). Newly free cells are depicted in green, newly occupied cells in red. The merging criteria are similar areas A_f, A_o, checked by $\frac{1}{A_{th}} < \frac{A_f}{A_o} < A_{th}$ with threshold A_{th}; similar angles ϕ_f, ϕ_o, checked by $\left(\min\left((\phi_f - \phi_o)\mathrm{mod}(\pi/2); \pi/2 - ((\phi_f - \phi_o)\mathrm{mod}(\pi/2))\right)\right) < \phi_{th}$ to account for the fact that the angles at this stage are ambiguous with respect to the quadrant and a merging can occur via the length or width of the rectangle; maximal distance d, checked by $d < d_{th}$; and a minimal percentage p of occupied cells within the connection line of the center of masses of the two rectangles, checked by $p > p_{th}$. The result of the merging in form of the blue, combined rectangle is shown in (c) while the final, orange rectangle that has been shortened by the newly free cells and that corresponds to the reconstruction of the original object is visualized in (d).

orientation ψ in the direction of newly occupied cells as well as length l and width w are measured and subsumed in the measurement vector $\boldsymbol{z} = (z_1, \dots, z_5)^T = (x, y, \psi, l, w)^T$. The set of all N_z measurements in time step k is given by $\mathcal{Z}_k = \{\boldsymbol{z}_{i,k}\}_{i=1}^{N_z}$, which are subsequently filtered by an IMM-UK-PDA tracker in order to reason about whether their movements are consistent with movement characteristics of real dynamic objects.

B. IMM-UK-PDA Tracking

For this purpose, we consider a model set $\mathcal{M} = \{M_i\}_{i=1}^{r}$ of $r = 3$ models within the tracking filter that share a common state vector $\boldsymbol{x} = (x, y, \psi, v, \omega, l, w)^T$ consisting of object center position within the grid map-fixed coordinate system x, y, heading angle ψ, velocity magnitude in driving direction v, turn rate ω as well as object length l and width w. Two of them model physically consistent motion behavior of real objects in form of a (nearly) Constant Velocity (CV) model M_1 given by the discrete-time stochastic state equation

$$\boldsymbol{x}_{k+1} = \begin{pmatrix} x_k + \frac{v_k T \sin\psi_k}{c} \\ y_k + \frac{v_k T \cos\psi_k}{c} \\ \psi_k \\ v_k \\ 0 \\ l_k \\ w_k \end{pmatrix} + \boldsymbol{u}_k + \boldsymbol{w}_k, \tag{14}$$

and for turning objects in form of a (nearly) Constant Turn Rate and Velocity (CTRV) model M_2 given by

$$\boldsymbol{x}_{k+1} = \begin{pmatrix} x_k + \frac{v_k}{c\omega_k}(\cos\psi_k - \cos(\omega_k T + \psi_k)) \\ y_k + \frac{v_k}{c\omega_k}(\sin(\omega_k T + \psi_k) - \sin\psi_k) \\ \psi_k + \omega_k T \\ v_k \\ \omega_k \\ l_k \\ w_k \end{pmatrix} + \boldsymbol{u}_k + \boldsymbol{w}_k. \tag{15}$$

considering a grid cell length c. Each share a common artificial input vector $\boldsymbol{u}_k$ that models the purely translational, deterministic grid map shift due to ego motion and have individual zero-mean, white, Gaussian noise sequences $\boldsymbol{w}_{1,k}, \boldsymbol{w}_{2,k}$ with corresponding covariance matrices $\boldsymbol{Q}_{1,k}, \boldsymbol{Q}_{2,k}$. These models are common within maneuvering multi-target tracking applications and are extensions of the ones found for example in [28]. We now supplement the models with a third model M_3 that corresponds to random movements, in fact being a stationary model, given by

$$\boldsymbol{x}_{k+1} = (x_k, y_k, \psi_k, 0, 0, l_k, w_k)^T + \boldsymbol{u}_k + \boldsymbol{w}_k \tag{16}$$

with much larger process noise covariance $\boldsymbol{Q}_{3,k}$. The common measurement equation is given by

$$\boldsymbol{z}_k = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \boldsymbol{x}_k + \boldsymbol{v}_k. \tag{17}$$

The aim now is to estimate the probabilities that temporally consecutive measurements correspond to real dynamic objects, therefore behaving according to a CV/CTRV model, or whether they are rather generated by random noise or static environment false positives and therefore better fitting the random movement model. The reason why the chosen model set allows this classification is further explained in Fig. 4 in a one-dimensional example.

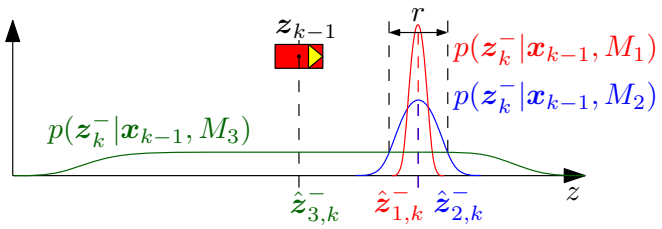

Fig. 4. We suppose the dynamic object is moving to the right. Its last position is given by $\boldsymbol{z}_{k-1}$ and the corresponding states $\boldsymbol{x}_{k-1}$ include a positive velocity. The measurement prediction densities for this scenario $p(\boldsymbol{z}_k^- | \boldsymbol{x}_{k-1}, M_j)$ are qualitatively visualized. All of the three models can describe any kind of movement theoretically. The difference is that they can do this to a different extent. The models M_1 and M_2 make sharper predictions centered at the predicted position further to the right while M_3 spreads its predicted probability over a much wider measurement space with minor overlap on the consistent models and remains centered at the old position. If a new measurement at time step k now falls in the measurement space r, the consistent models explain the measurement much better. This results in a higher mode probability for M_1 and M_2. The fact that the filter favors the model that fits better to the measurement sequence (higher mode probability) allows a classification of consistent and inconsistent movement behavior with negligible degradation of tracking performance. The more specific the prediction models, the better the discrimination capabilities.

Due to nonlinearities and clutter, the filtering is performed as explained in section II, so that states and mode probabilities are recursively estimated (Fig. 2g). However, instead of using the predicted measurement with the largest – in terms of determinant – innovation covariance $\boldsymbol{S}$, we only consider the model subset $\mathcal{M}_s = \{M_1, M_2\}$, so that j_r in (8) equals $j_r = \arg\max_{j \in \mathcal{M}_s} |\boldsymbol{S}_{j,k}|$. Otherwise, the large innovation covariance of the random movement model would amplify the track coalescence effect stemming from the associated large gates within a probabilistic data association. To prevent the problem of the discontinuity at the zero and 2π angle crossing within the tracking, the angle state is unbounded and measured angles with $\psi \in [0, 2\pi[$ are artificially increased by multiples of 2π to best fit the state prediction. This modification also solves the problem within the interaction step (3a) if one model already crossed the discontinuity and the others did not.

C. Classification and Grid Post Processing

For the decision if a cell within a dynamic rectangle object hypothesis can be classified as dynamic, the random movement mode probability μ_3 is compared to a threshold $\mu_{3,th}$ as soon as a track is confirmed. All cells that belong to tracks classified as dynamic are then artificially set to free in the actual grid map as the map shall contain only the static world and areas that other dynamic objects passed are considered static free space. For this purpose, bounding rectangles are calculated for each track with their actual refined

length and width estimations and rasterized on the grid by means of the Bresenham algorithm and under consideration of the past ego motion. A subsequent flood fill procedure is performed for clearing dynamic object mapping errors and the final, optimized grid map $\boldsymbol{m}_{\mathrm{opt},k}$ is generated (Fig. 2h). If the online classification decision of a cell being dynamic is not supported any more by a sufficiently low random movement mode probability, its actual occupancy probability is reinserted into the grid, which is considered a further advantage.

IV. Evaluation

The system is evaluated in 17 real traffic scenarios (2766 grid maps, 500000 cells per grid) that comprise 10 vehicle following and 7 crossroad scenarios in inner cities and freeways. Fig. 5 shows an excerpt of four scenes to get a better idea of the scenarios and achievable results. All scenarios contain a large amount of static environment to be mapped simultaneously such as parking cars, guardrails, trees that are partly in direct vicinity of dynamic vehicles. The parameters used within the evaluation can be found in Table I. The evaluation is performed on track level as well as on cell level. On track level, the system is in fact a pattern recognition system trying to classify emerging tracks as originating from real dynamic objects or not. Defining a True Positive (TP) classification as "*Real dynamic object correctly classified as real dynamic object*" and a False Positive (FP) as "*Non-dynamic object misclassified as real dynamic object*", it is obvious that FP's can have severe consequences for collision avoidance systems due to the fact that static environment cells would falsely be cleared from the grid. Therefore, a track already counts as FP, if it is misclassified as a real dynamic object at least once.[6] In contrast, a False Negative (FN) is not as severe, because in this case a real dynamic object and its cell corruptions will remain present in the grid as static obstacles. 518 tracks have been initialized in total. The classification results are 33 TP, 7 FP, 475 TN and 3 FN, resulting in a precision of 0.83, recall of 0.92 and a balanced accuracy of 0.95. The mean decision time for classifying a TP starting with the initial detection is (3.3 ± 0.57) time steps. This is equal to the timespan after which the grid cleaning procedure begins. The mean timespan for which the seldom FP classifications persist until falsely cleared cells are reinserted into the grid lies at only (3.5 ± 0.18) times steps. Consequently, each FP has only a very short-term negative effect. For the cell level classification, we examine how many grid cells are correctly cleared from dynamic object corruptions. For this purpose, all grids were manually labeled by marking areas that are completely free in reality, but not in the grid due to dynamic object corruptions. All cells in these areas ($> 2 \cdot 10^6$ in total) are checked of whether being successfully cleared by the algorithm or not. Fig. 6 shows the final result in form of a distribution of the number of remaining dynamic cell corruptions for different occupancy probability intervals compared to the standard grid mapping (left) as well as the percentage of successfully cleaned cells within each interval (right). In total, 75.5% of dynamic object cell corruptions were successfully removed with the proposed algorithm.

TABLE I

Parameters used in the experimental validation

Parameter	Value
$c; T; q_{th}$	0.16 m; 0.065 s; 0.9 ‰
$N_{DB}; r_{DB}$	12; 3 cells
$A_{th}; \phi_{th}$	3; 40 °
$d_{th}; p_{th}$	12 m; 20%
$\boldsymbol{Q}_1$	$\mathrm{diag}\big(0, 0, (5\,°/\mathrm{s})^2, (2\,\mathrm{m/s^2})^2, 0, (1\,\mathrm{m/s})^2, (0.5\,\mathrm{m/s})^2\big)T^2$
$\boldsymbol{Q}_2$	$\mathrm{diag}\big(0, 0, 0, (2\,\mathrm{m/s^2})^2, (50\,°/\mathrm{s^2})^2, \ldots$ $(1\,\mathrm{m/s})^2, (0.5\,\mathrm{m/s})^2\big)T^2$
$\boldsymbol{Q}_3$	$\mathrm{diag}\big((\frac{40}{c}\,\mathrm{m/s})^2, (\frac{40}{c}\,\mathrm{m/s})^2, (40\,°/\mathrm{s})^2, 0, 0, \ldots$ $(2\,\mathrm{m/s})^2, (1\,\mathrm{m/s})^2\big)T^2$
$\boldsymbol{R}$	$\mathrm{diag}\big((\frac{1}{c}\,\mathrm{m})^2, (\frac{1}{c}\,\mathrm{m})^2, (10\,°)^2, (0.34\,\mathrm{m})^2, (0.2\,\mathrm{m})^2\big)$
$\boldsymbol{\Pi}$	$p_{ij} = 0.96\ \forall i = j; \quad p_{ij} = 0.02\ \forall i \neq j$
$\boldsymbol{P}_{\{1,2,3\},0}$	$\mathrm{diag}\big((\frac{1}{c}\,\mathrm{m})^2, (\frac{1}{c}\,\mathrm{m})^2, (10\,°)^2, (10\,\mathrm{m/s})^2, (25\,°/\mathrm{s})^2, \ldots$ $(0.34\,\mathrm{m})^2, (0.2\,\mathrm{m})^2\big)$
$\boldsymbol{\mu}_0; \boldsymbol{x}_0$	$(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})^T; \quad (z_{1,0}, z_{2,0}, z_{3,0}, 0, 0, z_{4,0}, z_{5,0})^T$
$\alpha_U; \beta_U; \kappa_U$	0.001; 2; 0
$P_D; P_G$	0.7; 0.9
$\mu_{3,th}$	0.05

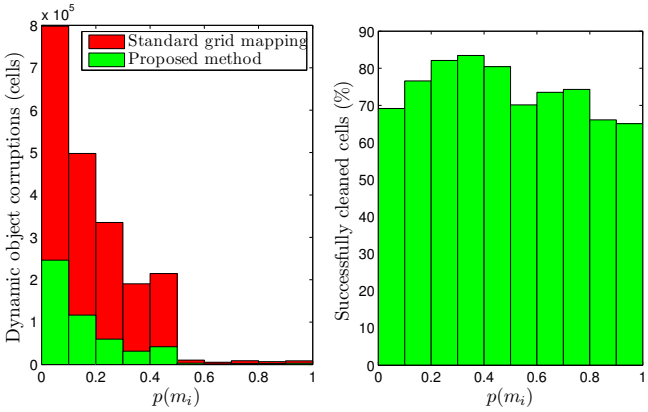

Fig. 6. Left: Number of dynamic object corruptions. Right: Percentage of successfully cleaned cells.

Acknowledgments

We kindly thank the Continental AG for cooperation and funding within the PRORETA 3 project, which aims at the development of future concepts for integrated driver assistance systems.

References

[1] R. Grewe, A. Hohm, S. Hegemann, S. Lueke *et al.*, "Towards a Generic and Efficient Environment Model for ADAS," in *Proc. of the IEEE Intelligent Vehicles Symposium*, Alcalá de Henares, Spain, Jun. 2012, pp. 316–321.

[2] M. Schreier, V. Willert, and J. Adamy, "From Grid Maps to Parametric Free Space Maps - A Highly Compact, Generic Environment Representation for ADAS," in *Proc. of the IEEE Intelligent Vehicles Symposium*, Gold Coast, Australia, Jun. 2013, pp. 938–944.

[3] E. Bauer, F. Lotz, M. Pfromm, M. Schreier *et al.*, "PRORETA 3: An Integrated Approach to Collision Avoidance and Vehicle Automation," *at - Automatisierungstechnik*, vol. 60, no. 12, pp. 755–765, Dec. 2012.

[6] Same for TP's. A track counts as TP, if correctly classified as dynamic at least once. If a TP result is falsely withdrawn over time, it will not alter the classification statistics, but have negative effects on the cell level evaluation.

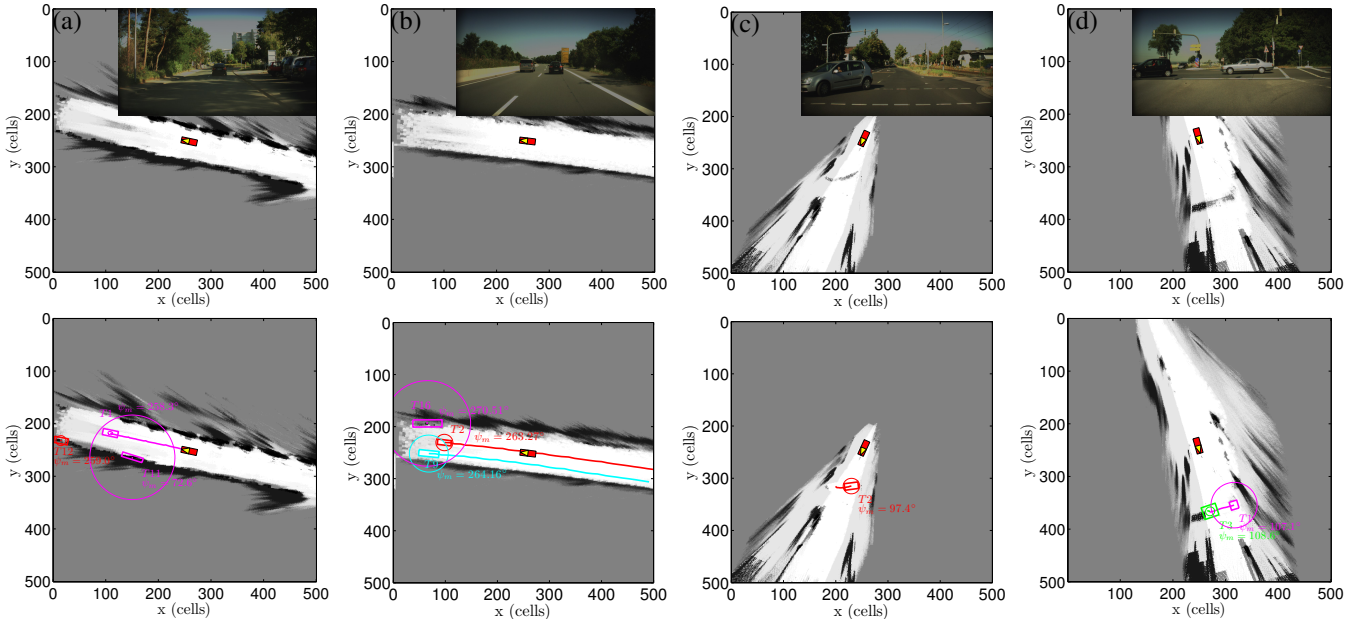

Fig. 5. Top: Original grid $\boldsymbol{m}_k$. Bottom: Optimized grid $\boldsymbol{m}_{\text{opt},k}$ with tracks. Only tracks classified as consistently moving are cleared from $\boldsymbol{m}_k$.

[4] M. Schreier and V. Willert, "Robust Free Space Detection in Occupancy Grid Maps by Methods of Image Analysis and Dynamic B-Spline Contour Tracking," in *Proc. of the IEEE International Conference on Intelligent Transportation Systems*, Anchorage, USA, Sep. 2012, pp. 514–521.

[5] V. Willert and J. Eggert, "A Stochastic Dynamical System for Optical Flow Estimation," in *Proc. of the 12th IEEE International Conference on Computer Vision, 4th Workshop on Dynamical Vision*, Kyoto, Japan, Sep. 2009, pp. 711–718.

[6] W. Burgard and M. Hebert, "World Modeling," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Berlin: Springer-Verlag, 2008, ch. 36, pp. 853–869.

[7] C. Coué, C. Pradalier, C. Laugier, T. Fraichard *et al.*, "Bayesian Occupancy Filtering for Multitarget Tracking: An Automotive Application," *The International Journal of Robotics Research*, vol. 25, no. 1, pp. 19–30, Jan. 2006.

[8] C. Chen, C. Tay, C. Laugier, and K. Mekhnacha, "Dynamic Environment Modeling with Gridmap: A Multiple-Object Tracking Application," in *Proc. of the International Conference on Control, Automation, Robotics and Vision*, Singapore, Dec. 2006, pp. 1–6.

[9] R. Danescu, F. Oniga, and S. Nedevschi, "Particle Grid Tracking System for Stereo Based Environment Perception," in *Proc. of the IEEE Intelligent Vehicles Symposium*, San Diego, CA, USA, Jun. 2010, pp. 987–992.

[10] D. F. Wolf and G. S. Sukhatme, "Mobile Robot Simultaneous Localization and Mapping in Dynamic Environments," *Autonomous Robots*, vol. 19, no. 1, pp. 53–65, Jul. 2005.

[11] C.-C. Wang, C. Thorpe, S. Thrun, M. Hebert *et al.*, "Simultaneous Localization, Mapping and Moving Object Tracking," *The International Journal of Robotics Research*, vol. 26, no. 9, pp. 889–916, Sep. 2007.

[12] D. Hähnel, D. Schulz, and W. Burgard, "Map Building with Mobile Robots in Populated Environments," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, Lausanne, Switzerland, Oct. 2002, pp. 496–501.

[13] T.-D. Vu, O. Aycard, and N. Appenrodt, "Online Localization and Mapping with Moving Object Tracking in Dynamic Outdoor Environments," in *Proc. of the IEEE Intelligent Vehicles Symposium*, Istanbul, Turkey, Jun. 2007, pp. 190–195.

[14] T.-D. Vu and O. Aycard, "Laser-based Detection and Tracking Moving Objects using Data-Driven Markov Chain Monte Carlo," in *Proc. of the IEEE International Conference on Robotics and Automation*, Kobe, Japan, May 2009, pp. 3800–3806.

[15] A. Petrovskaya and S. Thrun, "Model Based Vehicle Detection and Tracking for Autonomous Urban Driving," *Autonomous Robots*, vol. 26, no. 2-3, pp. 123–139, Apr. 2009.

[16] M. E. Bouzouraa and U. Hofmann, "Fusion of Occupancy Grid Mapping and Model Based Object Tracking for Driver Assistance Systems using Laser and Radar Sensors," in *Proc. of the IEEE Intelligent Vehicles Symposium*, San Diego, CA, USA, Jun. 2010, pp. 294–300.

[17] H. Lategahn, T. Graf, C. Hasberg, B. Kitt *et al.*, "Mapping in Dynamic Environments Using Stereo Vision," in *Proc. of the IEEE Intelligent Vehicles Symposium*, Baden-Baden, Germany, Jun. 2011, pp. 150–156.

[18] T.-N. Nguyen, M.-M. Meinecke, M. Tornow, and B. Michaelis, "Optimized Grid-Based Environment Perception in Advanced Driver Assistance Systems," in *Proc. of the IEEE Intelligent Vehicles Symposium*, Xi'an, Shaanxi, China, Jun. 2009, pp. 425–430.

[19] T.-N. Nguyen, B. Michaelis, A. Al-Hamadi, M. Tornow *et al.*, "Stereo-Camera-Based Urban Environment Perception Using Occupancy Grid and Object Tracking," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 1, pp. 154–165, Mar. 2012.

[20] A. Ess, K. Schindler, B. Leibe, and L. Van Gool, "Object Detection and Tracking for Autonomous Navigation in Dynamic Environments," *International Journal of Robotics Research*, vol. 29, no. 14, pp. 1707–1725, Dec. 2010.

[21] Y. Bar-Shalom and X.-R. Li, *Multitarget-Multisensor Tracking: Principles and Techniques*. Storrs, CT: Yaakov Bar-Shalom, 1995.

[22] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation. Theory, Algorithms and Software*. New York: Wiley-Interscience, 2001.

[23] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Boston: Artech House, 1999.

[24] X.-R. Li and V. P. Jilkov, "Survey of Maneuvering Target Tracking. Part V: Multiple-Model Methods," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1255–1321, Oct. 2005.

[25] S. J. Julier and J. K. Uhlmann, "A New Extension of the Kalman Filter to Nonlinear Systems," in *Proc. of SPIE3068: Signal Processing, Sensor Fusion, and Target Recognition VI*, vol. 3068, Orlando, FL, USA, Jul. 1997, pp. 182–193.

[26] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, ser. Intelligent Robotics and Autonomous Agents. Cambridge, Massachusetts: The MIT Press, 2006.

[27] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," in *Proc. of the International Conference on Knowledge Discovery and Data Mining*, Portland, Oregon, USA, Aug. 1996, pp. 226–231.

[28] X.-R. Li and V. P. Jilkov, "Survey of Maneuvering Target Tracking. Part I: Dynamic Models," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1333–1364, Oct. 2003.