OBJECT TRACKING AND DYNAMIC ESTIMATION ON EVIDENTIAL GRIDS

Ruben Jungnickel, *
Hamburg University of Applied Sciences, Dept. Computer Science, Berliner Tor 7
20099 Hamburg, Germany
ruben.jungnickel@haw-hamburg.de

ABSTRACT

Autonomous driving is one of the most challenging tasks of the automotive industry. As a subtask, the estimation of driveable and non driveable space is often solved by applying occupancy grids. The information about non driveable space can be used to improve object tracking. This paper presents an approach for object tracking and modelling in an occupancy grid map. Tracking objects on grid cells yields the advantage of a consistent environmental model on the occupancy grid map. We introduce the occupancy grid map as the only information source for the object tracking module. Taking advantage of the Dempster Shafer theory, a dynamic belief of conflicting cells can be estimated. This dynamic belief is then accumulated in a tracked object model. This is a grid based free form object model that uses an object local grid map representation to model vehicles in urban environment. We reduce false positives and initialization time by maintaining a dynamic belief for each object.

1. INTRODUCTION

An accurate environmental perception is a requirement for advanced driver assistant systems (ADAS) and mandatory for autonomous vehicles. As a modeling technique of the static environment, the occupancy grid framework proposed in [1] has dominated the scope and has been applied successfully to multiple systems [2] [3]. The original version of occupancy grid mapping suffers from artefacts produced by dynamic objects [4]. To prevent these issues, there are several techniques to filter out dynamic information or use them to set up grid map based tracking approaches. The basic idea of occupancy grid mapping is to model the environment as a set of discrete cells containing probabilities of the presence (occupancy) or absence (freeness) of an object. Elfes [1] proposes using a binary bayesian filter modeling the probability that a cell is occupied. In recent literature this idea has been extended by the Dempster Shafer theory (DST) using independent belief masses for occupancy and freeness. This enables the resolution of conflicts between independent contradictory measurements. Moras [5] presents an approach using an evidential dynamic detection based on the measured conflict of a cell. We extend this dynamic detection with an estimation that a cell is static using neighbouring cells. Representing a consistent environment model on grid maps including dynamic objects turns out to be a challenge. Modelling object dynamics in grid maps is described in several publications [5] [3] [2]. However there are still open questions how to use the extended information generated by a grid map in the object tracking later on.

Two basic techniques exist for modelling dynamics in occupancy grid maps.

- Cell attribute extension: extend each grid cell with dynamic information and track each of these cells individually [6].
- Associate cells to Tracks: associate a grid cell or cluster to a separated filter and keep track of the moving cells independently such as proposed in [3] [2].

Using the cell association technique for object tracking, information can be gathered in two ways [7].

- Extract features describing the tracked object from an occupancy grid map and transform these feature to vehicle coordinates. A feature can be the shape or contour of an object. The tracking uses the extracted features to update the object states.
- Use the cell representation of the occupancy grid map directly in object tracking. The object state is represented in grid coordinates and is updated by a set of cells.

In this paper we propose a novel approach using grid cells extended with dynamic belief masses to set up and validate new object tracks. Using these grid cells we associate multiple cells to a moving object employing the cell representation directly to update the object state. We set up particle filtered objects that accumulate dynamic measurements in a separated Dempster Shafer belief mass. These accumulated belief masses will determine if the target is generated from clutter or from a real moving object (false positive reduction). In some highway scenarios, the tracking suffers from uncertain distance measurements of the road boundary. In such scenarios we use dynamic beliefs for pruning ghost objects before they could produce failures in driver assistant systems. We show that dynamic cell tracking could be used in urban and highway scenarios (see section [2]). We use a particle filter to track a dynamic cell cluster and then conceptually detach these cells and put them into an object local grid. Each particle consists of a cluster of grid cells. Therefore a particle set allows free form modelling on multiple clusters. These clusters are accumulated in an object local grid which represents the shape of an object as a footprint.

This paper is structured as follows. Section [4] gives a brief overview about our occupancy grid mapping and feature extraction. In section [3] we present our particle filter based tracking and give an overview of the track management. Some experimental results are given in section [4] and finally in section [5] conclusions and future work are discussed.

* This work was supported by Ibeo Automotive Systems GmbH
2. OCCUPANCY GRID MAPPING AND DYNAMIC ESTIMATION

In our case a complete representation of the environment is built from the measurements of multiple multilayer lidar sensors. We accumulate these measurements using a grid based fusion technique in a single occupancy grid map using DST. At the end of this section we introduce an approach for grid based dynamic estimation based on the work of [5].

2.1. Dempster Shafer Theory on Occupancy Grid Maps

The Dempster Shafer theory is a mathematical theory allowing the combinations of evidences [6]. An evidence can be estimated by multiple sources and an own degree of belief. In DST all possible hypothesis of a system are defined by a set \( \Omega \) of mutually exclusive propositions. We define for occupancy grid mapping the set as \( \Omega = \{f, o\} \) where \( f \) is the state for a free and \( o \) is the state for an occupied cell. The elements of the powerset \( 2^\Omega = \{\emptyset, \{f\}, \{o\}, \{f, o\}\} \) can be used to represent the actual state of the system. DST uses mass functions \( m : 2^\Omega \to [0,1] \) assigning a belief mass to each element of the power set. Notice that the sum over all mass function is defined by: \( \sum_{A \subseteq \Omega} m(A) = 1 \).

\[
m_{M \otimes N}(A) = \frac{1}{\eta} \sum_{B \subseteq C \subseteq A \neq \emptyset} m_M(B) \cdot m_N(C) \quad (1)
\]

\[
\eta = 1 - \sum_{B \subseteq C = \emptyset} m_M(B) \cdot m_N(C)
\]

The set \( A \subseteq \Omega \) contains all elements that represent a state of interest. Dempster’s rule for combination (see equation 1) is used to combine the two sets of masses \( m_M \) and \( m_N \). Where \( \eta \) can be regarded as the agreement of the belief of the masses. The fusion process of an occupancy grid map is committed on cell level. Extending the idea of occupancy grid mapping, we create separated measurement occupancy grids for each purpose. Layer\(_1\) encompasses the original idea of grid mapping and builds a model for the static environment in the view of the ego vehicle. In Layer\(_2\) we use the power set \( 2^\Omega \). The measurement grid map and the occupancy grid map itself are both grid maps with the same attributes. So we are defining two mass functions: \( m_{S,k}(\cdot) \) as the mass function for the occupancy grid and \( m_{S,k}(\cdot) \) for the measurement grid.

2.2. Fusion Architecture

Accumulating information gathered from multiple sensors at different time slots a fusion process is needed in order to build a consistent grid map. For each laser scanner an own measurement grid map (MeasGrid) is built (see figure 1). So the process of occupancy grid mapping is triggered by every lidar scan. All measurement grids have to be in the same coordinates as the current grid map (GridMap\(_{k-1}\)) representation. Therefore each measurement has to be mapped to a joint coordinate system. Matching the measurement grid with the occupancy grid map (GridMap\(_{k-1}\)) the accurate vehicle position and orientation \( VehGridPos \) within the grid map has to be estimated. Using the estimated \( VehGridPos \) each measurement grid is fused with the occupancy grid map applying equation 4. The applied sensor model is split into two processes: the free space and the occupied space estimation. This separation is done because the free space estimation require a lot more sensor specific tuning. As long as similar laser scanners with the same update frequency are used, a measurement grid map can be fused to the occupancy grid map using Dempster’s rule of combination shown in equations 2...4.

\[
m_{M,k}(O) = m_{S,k-1 \odot S,k}(O) \quad (2)
\]

\[
m_{M,k}(F) = m_{S,k-1 \odot S,k}(F) \quad (3)
\]

\[
m_{S,k}(\Omega) = 1 - m_{S,k}(O) - m_{M,k}(F) \quad (4)
\]

The mass \( m_{S,k}(\cdot) \) is measured by an individual scanner \( S \) at time \( k \) \((F = \{f\}, O = \{o\})\).

2.3. Dynamic Estimation

A grid based dynamic detection is based on the assumption that a laser beam aimed at a static object should always end in the same cell. This assumption holds when an accurate vehicle odometry is available for map compensation. Using the occupancy grid map as a reference for all measurements accumulated until time slot \( k \) the current measurement can be matched against the map to enable two assumptions of the dynamic state: On the one hand a laser beam that ends in a free cell is an evidence of a static measurement. On the other hand a beam that ends in a free cell is an evidence of dynamic measurement (see equation 5).

The dynamic detection in occupancy grid maps is extensively studied with respects to the used grid mapping algorithm. Both [9] and [10] use conflicting measurements in bayesian occupancy grids. In contrast, [5] uses the same assumption of conflicting measurements in evidential grids. To validate the assumption of a dynamic measurement, it is useful to generate an estimation that a cell is not occupied by a dynamic object. These results can be integrated in a Dempster Shafer belief mass. For the dynamic estimation we define a second power set \( 2^D = \{\emptyset, \{d\}, \{\overline{d}\}, \{d, \overline{d}\}\} \), where \( d \) is the state of a dynamic measurement and \( \overline{d} \) is the state of a contradictory measurement. Representing dynamic measurements of the environment, we create a separate grid map layer Layer\(_3\) that includes all dynamic measurement. The belief masses \( m_{S,k}(D) \) and \( m_{S,k}(\overline{D}) \) describe a measurement of the dynamic and the not dynamic state \((D = \{d\}, \overline{D} = \{\overline{d}\})\). For the term \( m_{S,k}(D) \) the
motion detection algorithm proposed in [5] is used here. Equations 5 and 6 show the measurement of a conflict estimated to be dynamic as proposed in [5].

\[ C_1 = m_{S,k}(O) \cdot m_{M,k-1}(F), \text{ when } f \text{ turns to } o \]  
\[ C_2 = m_{S,k}(F) \cdot m_{M,k-1}(O), \text{ when } o \text{ turns to } f \]  
Where \( C_1 \) describes a cell which is currently occupied with a dynamic object (new dynamic). \( C_2 \) describes a cell which is freed by an moving object (old dynamic). Using the conflict \( C_1 \) between the measurement and the map, we extend this motion detection algorithm with an estimation \( D \), which determines that the current cell is not dynamic. Every measurement that falls on a free cell could be estimated as dynamic but due to discretization errors and noisy measurements we need an estimation that the cell is not dynamic \( D \). For every cell we take the state of neighbouring cells into account:

\[ m_{S,k}(D) = C_1 \]  
\[ m_{S,k}(D) = \alpha \sum_{n \in N} m_{M,n,k-1}(O) \]  
The mass belief function for \( m_{S,k}(D) \) can be described as the sum of all neighbour cells \((n \in N)\) where \( \alpha \) is a normalization constant and \( N \) is the set of relevant neighbours. We propose to take the belief of all eight neighbouring cells as the noise measurement sum for belief mass \( m_{S,k}(D) \). For the belief mass of \( m_{S,k}(D) \) we only take the conflict measurement \( C_1 \) into account, because \( C_1 \) describes where the moving object is currently located. Before we can combine the belief masses of Layer2, the movements of all object local grid maps have to be predicted. Therefore this update is separated from the static environment beliefs and is discussed in section 3.4.

3. DYNAMIC CLUSTER TRACKING

At this stage we have introduced a layered occupancy grid map built from several sensors. Layer1 for the static environment determine a cell is occupied or free and Layer2 for the dynamic environment estimating the belief for a dynamic and a not dynamic cell. In this chapter we only refer to Layer2 in order to track objects from the dynamic estimation of the grid map. To generate features from the dynamic estimation we build clusters for all dynamic cells using the db−scan algorithm [10]. Each cluster consists of a minimum of two cells each containing two beliefs: one for the dynamic and one for the not dynamic belief. Once a dynamic belief for each cell cluster is estimated, it can be used to set up object tracks. In contrast to other tracking approaches dynamic cell clusters, allow us to track the dynamic environment only.

The object model of a single tracked object is shown in figure 2. The track management, object model, association strategy and tracking algorithm are described in this section.

3.1. Particle Filters

A dynamic model and tracking algorithm is required for estimating directions and velocities. Cell clusters are a viable starting point. Vo [3] tracks dynamic clustered cells by using a Kalman filter and interactive multi model (IMM) algorithm. In contrast, [11] and [12] use particle filters. Particle filters belong to the monte carlo methods. They approximate the posterior density of the state space with a known transition model combined with a noise assumption. Or in other words particle filter approximate inference in partially observable Markov chains. Notice the state of the Markov chain at time \( t \) is given by \( x_t \). Furthermore, the state \( x_t \) depends on the previous state \( x_{t-1} \) according to the probabilistic law \( p(x_t|x_{t-1}) \), where \( u_t \) is the control arrived in the intervall \([t-1, t]\). The state in the Markov chain is not observable directly. But we can use the measurement \( z_t \), which can be used as a probabilistic projection to the true state \( x_t \) via the probabilistic law \( p(z_t|x_t) \). For the object tracking case, \( p(x_t|u_t, x_{t-1}) \) is referred as a motion model, and \( p(z_t|x_t) \) as the measurement model (see equation 7). Applying a particle filter to the object tracking problem the controls \( (u_t) \) are usually velocity and yaw-rate of the tracked object. The state of

![Figure 2: Object model in grid coordinates uses clustered dynamic cells (L). Grid map is rotated around ego vehicles course angle.](image)

**Algorithm 1: general particle filter algorithm see [13]**

**Data:** previous particles \( P_{f,t-1} \), control vector \( u_t \), measurement \( z_t \)

**Result:** particle weights \( w_k^i \), new particle set \( P_{f,t} \)

1. \( P_{f,t} = P_{f,t} = 0 \;
2. \text{for } m = 1 \ldots M \text{ particles in } P_{f,t-1} \text{ do}
3. \quad \text{sample } x_t^m \text{ with motion model } p(x_t|u_t, x_{t-1}^m) \;
4. \quad w_t^m = p(z_t|x_t^m) \;
5. \quad \text{// append calculated particle to set: } P_{f,t} = P_{f,t} + \{x_t^m, w_t^m\} \;
6. \text{end}
7. \text{for } i = 1 \ldots M \text{ particles in } P_{f,t} \text{ do}
8. \quad \text{draw } i \text{ with probability } \propto w_i^m \;
9. \quad \text{add } x_t^i \text{ to } P_{f,t} \;
10. \text{end}
11. \text{return } P_{f,t};
applied by drawing a sample in the state space with the probability
\[ p(x_t | u_t, x_{t-1}^n) \]. After the prediction step the particle is weighted
(line 4). The particle weight is determined by the likelihood of the agree-ment \( p(z_t | x_t) \) between the particle state and current measure-ment \( z_t \). The object state of a particle filter is strongly modi-fied by the weights of its particles (see [13] lines 8,9 ). Particle with
low weights are replaced with high weighted particles. This is the
typical survival of the fittest methodology of a particle filter.
As an advantage over other tracking algorithms, particle filters are
able to track non-linear state spaces and noise can be modeled in
any required form. We use particle filters as an experimental ap-proach because they are easy to implement and offer the ability to
track multiple object hypothesis in one filter.

3.2. Object Model
Taking advantage of the clustered dynamic cells proposed in the
last section, particle filters are used to represent an object. Parti-
cles that track cells as proposed in [11] [12] have the property of
converging on the measurement. This is no disadvantage as long
as the whole object is visible in the majority of measurements,
but it suffers when objects are partly occluded [13]. A box represen-
tation of the object to be tracked is used by [15] [16]. In an
object box representation all measurements have to be fitted to a
box. But a box might not be a good model for every situation. The
state of a particle is proposed as \( s(k) = (L, x, y, \dot{x}, \dot{y}, \omega) \) where
\( L = \{ (c_1), (c_2), \ldots, (c_n) \} \) is the set of clustered dynamic cells
describing the associated measurement (see figure 2).
We use a Constant Turn Rate and Velocity (CTRV) model as state
representation assuming the velocity \( (\dot{x}, \dot{y}) \) and the yaw rate \( \omega \) are
constant. The weight of a particle is estimated by the matching of
the particle’s object hypothesis and current measurements.

\[ p(z_t | x_t^p) = \lambda \cdot |L_{x_t^p} \cap L_{z_t^f}| \]  

(9)

The above equation is a matching function calculating the cluster
\( L \) overlap of the object hypothesis of a particle \( x_t^p \) and a measure-
ment \( z_t^f \). This weighting function could be described as
sum over all dynamic beliefs of the current measurement grid cell,
which matches with the cell cluster set \( L \) of the particle, normal-
ized by the constant \( \lambda \).
We combine all particles representing the same object in a parti-
cle set \( P_t \) in order to build a free form object model \( \sum_{p \in P_t} L \) (footprint in figure 3). Particles are initialized by a single dy-
amic cluster \( L \). A motion prediction step is performed in order to
match the position of the particles to the new measurement. If all
particles of a particle set are combined, they can be used as a free
form model, namely footprint \( F_p \) (see figure 3). The footprint
combines all particles with its cell cluster \( L \) to an object repre-
sentation in grid coordinates. The resulting object state is a set of
cells described by a hit counter. This hit counter is incremented
if the center of a tracked cell fits into an object local cell. Object
local grids were proposed first by [17]. We modify this approach
if the center of a tracked cell fits into an object local cell. Object
cells described by a hit counter. This hit counter is incremented
of this paper.
A new particle set is set up if too few particles exist in the associa-
tion region. To prevent false positives, the cell cluster used for
initialization has to pass the constraint \( B_{PD} \geq B_{PF} \), where \( (B_{PD}) \) is
the mean dynamic belief. As a first guess, all particles are initia-
Ized with the original dynamic cluster and a velocity in the interval
\([V_x, -V_x]\]. Only velocities within this interval can be tracked.

3.3.1. Growing
A particle filter once initialized by one dynamic cluster may de-
scribe multiple cluster representations. So new clusters should be
inserted if the original shape of dynamic measurements does not
match any more. Since we do not want to modify existing parti-
cles, we propose to add new particles to the particle filter. There-
fore each particle filter has a maximum size \( |P_{max}| \) but in the ini-
tial state all particle filters are able to grow or shrink. If a newly
added cluster matches well it will replace other particles, which do
not match anymore. That is a result of the survival of the fittest
methodology of the used SIR particle filter (only the best particle
are drawn). In [13] a similar aproach for growing and shrinking
of this paper.
A new particle set is set up if too few particles exist in the associa-
tion region. To prevent false positives, the cell cluster used for
initialization has to pass the constraint \( B_{PD} \geq B_{PF} \), where \( (B_{PD}) \) is
the mean dynamic belief. As a first guess, all particles are initia-
Ized with the original dynamic cluster and a velocity in the interval
\([V_x, -V_x]\]. Only velocities within this interval can be tracked.

3.3. Track Management
The proposed tracking approach encompasses the following track
management features:

- initialization of tracks using clustered dynamic cells

- growing and shrinking of particle sets (i.e. adding or re-
  moving new cluster estimations to the set).
- deletion of divergent particle sets

Fusion of two tracks tracking the same object is beyond the scope
of this paper.

Figure 3: sequential build of free form object model using parti-
cles. In t=2 new particles p4 and p5 are inserted to enable the filter
to track the new object shape.

- growing and shrinking of particle sets (i.e. adding or re-
  moving new cluster estimations to the set).
- deletion of divergent particle sets

Fusion of two tracks tracking the same object is beyond the scope
of this paper.

A new particle set is set up if too few particles exist in the associa-
tion region. To prevent false positives, the cell cluster used for
initialization has to pass the constraint \( B_{PD} \geq B_{PF} \), where \( (B_{PD}) \) is
the mean dynamic belief. As a first guess, all particles are initia-
Ized with the original dynamic cluster and a velocity in the interval
\([V_x, -V_x]\]. Only velocities within this interval can be tracked.

3.3.1. Growing
A particle filter once initialized by one dynamic cluster may de-
scribe multiple cluster representations. So new clusters should be
inserted if the original shape of dynamic measurements does not
match any more. Since we do not want to modify existing parti-
cles, we propose to add new particles to the particle filter. There-
fore each particle filter has a maximum size \( |P_{max}| \) but in the ini-
tial state all particle filters are able to grow or shrink. If a newly
added cluster matches well it will replace other particles, which do
not match anymore. That is a result of the survival of the fittest
methodology of the used SIR particle filter (only the best particle
are drawn). In [13] a similar aproach for growing and shrinking
of this paper.

A new particle set is set up if too few particles exist in the associa-
tion region. To prevent false positives, the cell cluster used for
initialization has to pass the constraint \( B_{PD} \geq B_{PF} \), where \( (B_{PD}) \) is
the mean dynamic belief. As a first guess, all particles are initia-
Ized with the original dynamic cluster and a velocity in the interval
\([V_x, -V_x]\). Only velocities within this interval can be tracked.

3.3. Track Management
The proposed tracking approach encompasses the following track
management features:

- initialization of tracks using clustered dynamic cells

- growing and shrinking of particle sets (i.e. adding or re-
  moving new cluster estimations to the set).
- deletion of divergent particle sets

Fusion of two tracks tracking the same object is beyond the scope
of this paper.

A new particle set is set up if too few particles exist in the associa-
tion region. To prevent false positives, the cell cluster used for
initialization has to pass the constraint \( B_{PD} \geq B_{PF} \), where \( (B_{PD}) \) is
the mean dynamic belief. As a first guess, all particles are initia-
Ized with the original dynamic cluster and a velocity in the interval
\([V_x, -V_x]\). Only velocities within this interval can be tracked.

3.3.1. Growing
A particle filter once initialized by one dynamic cluster may de-
scribe multiple cluster representations. So new clusters should be
inserted if the original shape of dynamic measurements does not
match any more. Since we do not want to modify existing parti-
cles, we propose to add new particles to the particle filter. There-
fore each particle filter has a maximum size \( |P_{max}| \) but in the ini-
tial state all particle filters are able to grow or shrink. If a newly
added cluster matches well it will replace other particles, which do
not match anymore. That is a result of the survival of the fittest
methodology of the used SIR particle filter (only the best particle
are drawn). In [13] a similar aproach for growing and shrinking
of this paper.

A new particle set is set up if too few particles exist in the associa-
tion region. To prevent false positives, the cell cluster used for
initialization has to pass the constraint \( B_{PD} \geq B_{PF} \), where \( (B_{PD}) \) is
the mean dynamic belief. As a first guess, all particles are initia-
Ized with the original dynamic cluster and a velocity in the interval
\([V_x, -V_x]\). Only velocities within this interval can be tracked.

3.3. Track Management
The proposed tracking approach encompasses the following track
management features:

- initialization of tracks using clustered dynamic cells

- growing and shrinking of particle sets (i.e. adding or re-
  moving new cluster estimations to the set).
- deletion of divergent particle sets

Fusion of two tracks tracking the same object is beyond the scope
of this paper.

A new particle set is set up if too few particles exist in the associa-
tion region. To prevent false positives, the cell cluster used for
initialization has to pass the constraint \( B_{PD} \geq B_{PF} \), where \( (B_{PD}) \) is
the mean dynamic belief. As a first guess, all particles are initia-
Ized with the original dynamic cluster and a velocity in the interval
\([V_x, -V_x]\). Only velocities within this interval can be tracked.
A new measurement should only be added to the set if it matches the track well $\beta \geq l_s$. Where $\beta$ is the assignment probability that a track matches the measurement. This constraint avoids that wrong associations can break the object state.

3.3.2. Shrinking

In contrast to the growing process the shrinking process only needs to be called once for each particle filter. We define a minimal $|P_{\text{meas}}|$ and maximal $|P_{\text{max}}|$ amount of particles for each particle set. $|C_{\text{meas}}|$ is the cluster size of the measured cluster associated with the track. Let $|T|$ be the mean cluster size of the particle set $P_T$. The assignment probability $\beta$ describes the likelihood that $P_T$ is associated with $C_{\text{meas}}$. The shrinking of a particle set is triggered if the association probability is less than a certain threshold $l_s$. Additionally the growing of the particle filter is governed by the threshold $l_g$.

Recall that the positional covariance matrix $C_{P_{\text{meas}}}$ describes an ellipse which expresses the shape of the corresponding point cloud. Since shrinking is applied after all measurements have been associated, we have to compare against the eccentricity $C_{P_{\text{meas}}}$. The amount of particles in a particle set $|P_T|$ is changed adaptively with equation (10). The growing / shrinking function is applied at the weighting step of each particle filter.

3.3.3. Deletion

If any tracked object is not observable anymore, the particle set should be deleted. The deletion can be triggered by two conditions as listed in Table 1. A particle filter should be deleted if the position of the tracked object $P_{fP_{\text{meas}}}$ is not in the perception area of the map $M$. A second condition for deletion of an established track is passed if the filter has lost track of an object. This behaviour can be determined easily if the particle filter diverges.

3.4. Accumulating Dynamic Estimation for each Object

Validation of resulting tracked objects can be achieved by accumulating the dynamic belief of a measurement ($m_{O,k}(D)$) in a separate belief mass determined by the dynamic belief of an object. The belief mass $m_{O,k}(D)$ is stored as an object attribute. The dynamic belief for an object is updated using the associated measurement ($m_{S,k}$) of a cluster. We use the average of the dynamic beliefs of all relevant cells as the dynamic belief of a cluster.

\[
\begin{align*}
    m_O(D) &= m_{O,k-1}\oplus S_k(D) \\
    m_{O,k}(D) &= m_{O,k-1}\oplus S_k(T)
\end{align*}
\]

Using Dempster’s rule again to combine the measured mass with the prior dynamic estimation, a dynamic belief for each object can be maintained.

4. RESULTS

The proposed tracking algorithm and dynamic accumulation was tested on a vehicle equipped with two Ibeo LUX laser scanners.

The environmental modeling using grid maps and the proposed tracking algorithm are illustrated for two different scenarios. Figure 4a shows an urban environment with an ego velocity about 30 km/h. Figure 4b shows a highway scenario with ego velocities about 160 km/h. The Dempster Shafer belief masses are shown with the colours green for free, red for occupied and blue for the unknown belief mass (see figure 4c and 4d). The dynamic measurement (yellow) shown in figure 4b and 4c is strongly dependent on the estimated free space. With a higher belief of the free mass the estimation of the dynamic measurement is growing proportionally. This behaviour has the benefit that object close to the ego vehicle gain a higher dynamic belief. The colour purple shows the belief of a not dynamic measurement.

The red cluster (in figure 4c and 4d) represents the tracked object footprint. The yellow line describes the velocity vector of the vehicles. The footprint of the overtaking vehicle in 4c consist of all shapes ever seen by one scanner. The initial shape of the object is only the side of the vehicle. Using the proposed model, the object shape grows with new measurements (if the back of the car is seen). The proposed particle filter is able to track any shape of an object.

In figure 4c the dynamic estimation of the right road boundary appear dynamic (yellow). This is caused by a misaligned mounting position of the right laser scanner during the test drive. The algorithm was tested offline on an Intel Core i7 with 2.8 GHz. The map generation calculation time is about 3 ms. The tracking algorithm including clustering and some debug outputs is about 10 ms.

4.1. Implementation Details

The cell size of the used occupancy grid is 20 cm and we built the occupancy grid of size of 102.4 m x 102.4 m. We statically allocate memory for all particles ever used in the algorithm up to a maximum of 8000 particles. Each particle filter starts with 375 particles ($|P_{\text{min}}|$) and is able to grow to a size of 500 particles ($|P_{\text{max}}|$).

5. CONCLUSION AND FUTURE WORK

In this paper an environmental modeling technique for vehicles equipped with multiple laser scanners is presented. We propose to build a layered occupancy grid map to extend the model of a static environment with information produced by dynamic objects. These dynamic information could be clustered and tracked using particle filters. As free form object model, multiple particles generate an object local grid map. As mentioned in 5 the two types of cell conflict $C_1$ and $C_2$ could be used for estimating the direction of the motion. A direction could be useful for a faster track initialization of the used particle filter algorithm. The initial state of each particle could be initialized by a first estimation of the direction in order to improve the convergence speed of the particle filter.

We plan to compare the implemented particle filter tracking against a Kalman filter tracking comparing initialization time and position error.
Figure 4: Grid based tracking and dynamic detection in an urban and highway scenario. The dynamic measurement grid shows yellow for the dynamic belief mass and purple for the not dynamic belief mass.

6. ACKNOWLEDGEMENTS

This work is funded by the Federal Ministry of Education and Research of Germany (BMBF) within the RECBAR project.

7. REFERENCES


tenzsysteme mit hierarchischen Belegungskarten*, Ph.D. the-
sis, Universität der Bundeswehr München, Fakultät für Luft-
und Raumfahrttechnik, Neubiberg, 2012.

tracking from a mobile robot,” *International Journal of So-

[13] Sebastian Thrun, Wolfram Burgard, and Dieter Fox, *Prob-
abilistic Robotics*, Intelligent Robotics and Autonomous

multiple moving targets with a mobile robot using particle
filters and statistical data association,” in *Robotics and Au-

[15] Anna Petrovskaya and Sebastian Thrun, “Model based ve-
hicle detection and tracking for autonomous urban driving,”

[16] K. Schueler, T. Weiherer, E. Bouzouraa, and U. Hofmann,
“360 degree multi sensor fusion for static and dynamic ob-
stacles,” in *Intelligent Vehicles Symposium (IV), 2012 IEEE,

[17] Jan Effertz, *Autonome Fahrzeugführung in urbaner Umge-
bung durch Kombination objekt- und kartenbasieter Um-
feldmodelle*, Ph.D. thesis, Technischen Universität Carolo-
Wilhelmina zu Braunschweig, Februar 2009.