Proceedings of the 16th International IEEE Annual Conference on Intelligent Transportation Systems (ITSC 2013), The Hague, The Netherlands, October 6-9, 2013

An interval based representation of occupancy information for driver assistance systems

Tobias Weiherer, Sayed Bouzouraa and Ulrich Hofmann

Abstract— **Future Advanced Driver Assistance Systems (ADAS) require detailed information about occupancy states in the vehicle's local environment. In contrast to widespread occupancy grids, this information should be represented in a compact, scalable and easy-to-interpret data structure. In this paper, we show how occupancy probabilities can efficiently be represented in our 2D Interval Map framework. The basic idea of this approach is to discretize the vehicle's environment only in longitudinal direction and to avoid quantization errors in lateral direction by storing continuous values. In order to correctly deal with dynamic obstacles in ADAS scenarios, the map also interacts with a model based object tracking.**

The comparison of our experimental results to a ground truth illustrates the differences of grid and interval based environment representations. A tested collision avoidance function yields similar results for both representations, while computation times and memory requirements are substantially improved by the application of the 2D Interval Map.

I. Introduction

A. Motivation

The perception of the own vehicle's environment is one of the key challenges in the development of future Advanced Driver Assistance Systems (ADAS). This task includes the processing of uncertain sensor measurements in a generic environment representation so that it can be used by different driver assistance functions. In general, we categorize environment representations into map based approaches and model based object representations [1]. The latter make assumptions about common object properties, for example the shape and motion of dynamic obstacles. In contrast, map based representations assign location-specific information to dense space elements and provide a proximity relationship between these elements.

One crucial information that has to be represented for almost every highly automated vehicle is the knowledge about free spaces and obstacles in the system's local environment. Especially for mobile robots, this problem has been examined extensively over the last years, for example by using well-known occupancy grids [2].

In our work we focus on the development of ADAS for mostly longitudinal traffic in complex scenarios on highways, e.g. traffic jams. Compared to typical mobile robot environments, map based representations for automotive systems have to deal with a highly dynamic environment and have to consider a larger area around the own vehicle. Furthermore, the limited memory and computing capabilities of automotive electronic control units demand representations that are easy to interpret and very efficient concerning memory and computation requirements. For this purpose, today's common grid maps have to be further simplified in order to achieve a compact representation of environment information while still providing sufficient accuracy. The 2D Interval Map as a new generic map based environment representation and its application for convoy track detection have already been described in [3]. The contribution of this paper is twofold: First, we show how occupancy information can be represented in the 2D Interval Map data structure. Second, the interaction of the resulting occupancy map with a model based object tracking is introduced, which allows handling the highly dynamic environment in ADAS scenarios and generates a consistent environment representation.

B. Related Work

Map based environment representations are typically known from 2D occupancy grid maps in the field of robotics [2]. Besides 2D approaches, there exist numerous approaches to extend the concept of grid maps by a third dimension, for example voxel maps [1] or multi volume occupancy grids [4]. Moreover, grid maps can also represent different types of information, for example gray values of video images [5].

In general, grid based methods are confronted with comparatively high memory consumptions and computational effort. Moreover, they are not scalable and provide data in the same high accuracy in the whole map area, which is not necessary for most driver assistance functions. One possibility of scalable environment representations is given by tree based methods, for example [6], which are on the downside faced with rather high computational effort. Besides that, there are several feature and graph based approaches, for example [7], which have shortcomings due to restricting model assumptions about the environment. Another category of publications deals with the further processing of the information represented in occupancy grids, e.g. a compact representation of free spaces [8] or the compression of grid map data [9]. In contrast, this paper aims at already simplifying the data structure that is used for the accumulation process.

On the other hand, several works address typical challenges of grid maps in the automotive application area, for example the fusion of several different sensors [10] or the handling of dynamic objects. [11] and [12] use the results

T. Weiherer is with the Faculty of Electrical Engineering, Lehrstuhl fuer Datenverarbeitung, Technische Universitaet Muenchen, Germany weiherer@tum.de

S. Bouzouraa and U. Hofmann are with the Advanced Driver Assistance Development Group, AUDI AG, Germany essayed.bouzouraa@audi.de, ulrich.hofmann@audi.de

978-1-4799-2914-613/$31.00 ©2013 IEEE

of a model based object tracking to represent separated map layers for static and dynamic obstacles. Our concept of combining the results of map and object based environment representation by correctly describing dynamic objects in an occupancy map has also been presented [1], [13].

C. Structure of the paper

The rest of this paper is structured as follows: Section II gives an overview over our system architecture, section III introduces the general 2D Interval Map framework. The application of the 2D Interval Map for representing occupancy information is shown in chapter IV, the enhancements that are necessary to consider the motion of dynamic obstacles are presented in section V. Section VI analyzes experimental results, while section VII finally concludes the paper.

II. System Architecture

Figure 1 illustrates the two main philosophies of our system architecture.

First, we consider our perception system as a combination of experts with different capabilities. Figure 1 shows a combination of three experts: The 2D Interval Map as a map based environment representation, the object module as a model based object representation and the ego motion module, which is responsible for ego motion estimation. All participating subsystems closely interact in case they need information that is better provided by any other subsystem. The interaction of the 2D Interval Map and object tracking will be presented in section V. The combination of all available subsystems provides a consistent environment representation.

Second, our architecture is based on a separation of environment representations from driver assistance functions. As shown for the 2D Interval Map, sensor models are used to transfer measurements including uncertainties into the representation. Afterwards, multiple extractors can be deployed to extract information which is relevant for certain ADAS functions. This decoupling simplifies the development process of assistance systems.

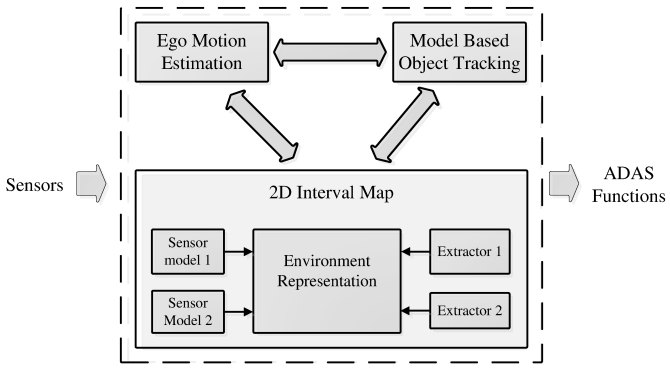

Fig. 1. Perception system architecture

III. 2D Interval Map

In contrast to grid based maps, the 2D Interval Map discretizes the space around the own vehicle only in longitudinal vehicle direction, whereas the lateral component of any information is stored as a continuous value, as already introduced in [3]. The main reason for this approach is an analysis

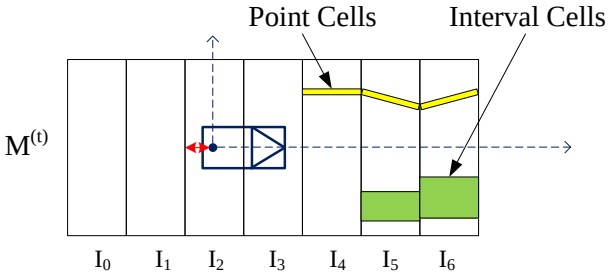

Fig. 2. 2D Interval Map data structure

showing that typical ADAS functions for longitudinal traffic demand a higher accuracy in lateral than in longitudinal direction. Another advantage lies in the fact that the map is always aligned in the same direction as the own vehicle's current heading, which makes the stored information easy-to-interpret. Fig. 2 illustrates an interval map $M^{(t)}$ at timestamp t. In each map, the vehicle's environment is partitioned into a static number of equally spaced **intervals** I:

$$M^{(t)} = \left[I_0^{(t)}, \ldots, I_n^{(t)}\right]^T \tag{1}$$

In this structure, each interval can store several generic information containers of different types. As also shown in Fig. 2, we subdivide these containers into punctual information, called **point cells** (PC) and information with dimension, called **interval cells** (IC):

$$I^{(t)} = \left[PC_0^{(t)}, \ldots, PC_k^{(t)}, IC_0^{(t)}, \ldots, IC_l^{(t)}, \ldots\right] \tag{2}$$

Both PC and IC data containers can include arbitrary information, yet they have to store at least one respectively two lateral positions:

$$PC^{(t)} = \left[y^{(t)}, \ldots\right] \tag{3}$$

$$IC^{(t)} = \left[y_{start}^{(t)}, y_{end}^{(t)}, \ldots\right] \tag{4}$$

If the content of point or interval cells is filtered by recursive estimation mechanisms, the attributes can be extended by corresponding covariance values.

As mentioned above, the 2D Interval Map is always aligned in the same direction as the own vehicle. Therefore, stored information has to be compensated to enable accumulation mechanisms in the presence of own vehicle motion. We already showed in [3] that the longitudinal and lateral motion components can be compensated exactly, whereas rotations have to be approximated. The detailed compensation implementation used in the 2D Interval Occupancy Map will be shown in section IV-A.

IV. 2D Interval Occupancy Map

Similar to occupancy grids, the main task of the 2D Interval Occupancy Map is to describe occupied, free and unknown areas around the own vehicle. As all these information

978-1-4799-2914-613/$31.00 ©2013 IEEE

have a dimension, they can be represented with interval cells in the 2D Interval Map framework. These interval cells can be modeled in different ways, which significantly impacts the further implementation of the map.

First, the cells of an interval can be either managed in a single list or in separated lists for each occupancy type, as for example used for the three dimensional multi volumes presented in [4]. One obvious drawback of keeping several lists is the occurrence of conflicting information and blank areas, which are not described by any stored cell. Furthermore, even if two neighboring cells of different types border on each other, the redundant lateral position of the border has to be stored twice. Therefore, our implementation is based on a dense list of interval cells, which only contains the lateral end position $y_{end}^{(t)}$ of a cell.

Second, the stored free and occupied cells have to model a reliability and hence enable an accumulation process. Regarding this issue, interval cells storing a common posterior occupancy probability offer the advantage that cells can change their state over time. Furthermore, well-known state estimation mechanisms can be adapted straightforwardly. Besides calculating the posterior occupancy probability of a cell $P\left(O|z^{(1:t)}\right)$ given all measurements $z^{(1:t)}$ until time t, we also estimate the cells' borders $y_{end}^{(t)}$ by using a one-dimensional Kalman-Filter. In this way, a border between two cells can be reused for slightly different occupancy measurements and is able to represent a smooth transition between occupancy states. The resulting estimation variance $\sigma^2\left(y_{end}\right)^{(t)}$ can be interpreted as a measure of smoothness, as also illustrated in figure 3.

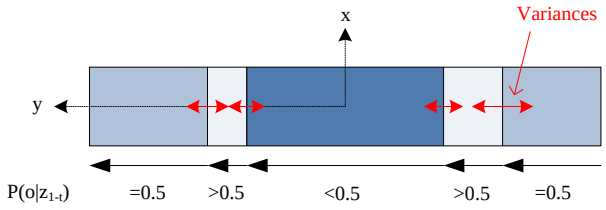

Fig. 3. Interval cell modeling

On this basis, the occupancy interval cells contain the following attributes:

$$IC^{(t)} = \Big[y_{end}^{(t)}, \sigma^2\left(y_{end}\right)^{(t)}, x^{(t)}, P\left(o|z^{(1:t)}\right), n_{age}^{(t)}, ID_{object}^{(t)}\Big] \quad (5)$$

where $n_{age}^{(t)}$ denotes the age of a cell and $ID_{object}^{(t)}$ the ID of an associated dynamic object. The usage of the lateral position $x^{(t)}$ will be presented in the next section. As only the lateral ends of all cells are incorporated, the start of the first cell in each interval has to be stored separately.

The different steps that are necessary to update the 2D Interval Occupancy Map with new sensor measurements are illustrated in figure 4 and will be explained in detail in the following sections. At the beginning, all intervals of the map are initialized with a single cell containing the a priori occupancy probability and an initial border variance. The presented approach can be combined with several sensor technologies, however, we focus on the processing of laser scanner measurements in this paper.

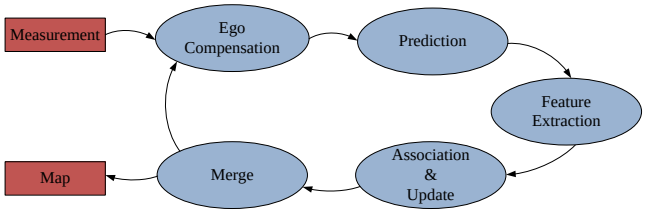

Fig. 4. 2D Occupancy Interval Map update cycle

A. Compensation of own vehicle motion

The first step in processing any local map based environment representation is to compensate the stored information by the own vehicle's motion between two map updates. This motion can be subdivided into three components. The longitudinal motion of the own vehicle can be accomplished by shifting the complete data structure and accumulating a sub-interval value. This processing has been described in detail in [3]. The lateral shift between two map updates is directly compensated by adjusting the interval cells' y-attributes. In order to provide a consistent data structure, a fixed width of the map can be defined. If the shifted cells exceed this width, they will be cropped, blank areas are filled with cells containing the a priori probability.

For compensating rotations of the own vehicle between two updates, all interval cells include a single approximated x-center value. By using this value, an absolute lever for each interval cell can be calculated. Afterwards, this lever is rotated in order to calculate the longitudinal and lateral difference resulting from the rotation. While the lateral component is again compensated directly, the longitudinal component is used to correct the cell's x-center. If this value exceeds the longitudinal borders, the cell has to be transfered to the neighboring interval.

B. Prediction of interval cell states

The estimation of the cells' borders by a Kalman filter requires a dedicated prediction step before the actual update takes place. For these borders we use a simplistic one-dimensional process and observation model without further input:

$$y_{end}^{(t)} = y_{end}^{(t-1)} + w^{(t-1)} \quad (6)$$

$$z_{end}^{(t)} = y_{end}^{(t)} + v^{(t)} \quad (7)$$

where both process noise $w^{(t)} \sim \mathcal{N}\left(0, \sigma_u^2\right)$ and observation noise $v^{(t+1)} \sim \mathcal{N}\left(0, \sigma_z^2\right)$ are Gaussian and zero-mean. $z_{end}^{(t)}$ denotes the measurement of the border at timestamp t.

In this case, the prediction step simplifies to:

$$\bar{\sigma}^2\left(y_{end}\right)^{(t)} = \sigma^2\left(y_{end}\right)^{(t-1)} + \sigma_u^2 \quad (8)$$

C. Feature Extraction

In order to update the cells' borders and occupancy probabilities, information about free and occupied areas has to be extracted from the laser range measurements. As illustrated in figure 5, we use the following rules to generate new cells from the laser measurements within an interval:

- Occupied cells are built from laser beam reflections. The dimensions of the occupied interval cells are calculated by considering the divergence of the laser beam and the longitudinal uncertainty of the measurement. Overlapping cells are merged to enable a consistent data structure.
- Free cells are built from the areas the laser beams passed. They are limited by unknown and occupied measurements. Again, the divergence of all involved beams has to be considered.
- The remaining areas are filled with cells containing the a priori probability in order to provide a dense data structure.

All extracted cells contain an occupancy probability given the current measurement. This probability $P\left(o|z_t\right)$ corresponds to the inverse sensor model, which is used in well-known occupancy grid update algorithms [2]. In our implementation, this probability is approximated for each generated cell, depending on the number and arrangement of the laser beams as well as the overall distance of the laser range measurements. Besides that, the characteristics of the laser measurements at the transitions of occupancy states could be used to approximate the variance σ_z^2 of a border measurement.

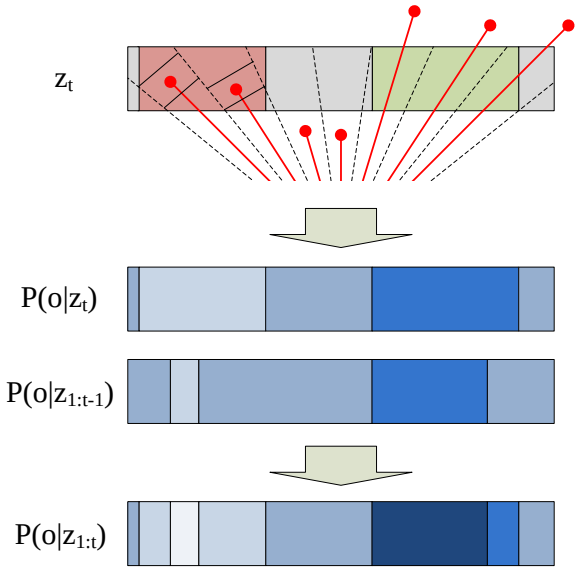

Fig. 5. Feature extraction and update of interval cells. Darker colors indicate low occupancy probabilities, bright colors high occupancy probabilities

D. Association and Update

Having extracted cells from the laser range measurements, the map interval structure can be updated. Therefore, both introduced state estimation mechanisms have to be considered: On the one hand, the inverse sensor model probabilities have to be assigned to the existing cells. On the other hand, existing and detected borders have to be associated and updated. If a newly detected border cannot be associated to an existing one, a new cell has to be established. The complete association and update algorithm of the cell data structure is summarized in the pseudo code in figure 6 and also visualized in figure 5.

```
1:  procedure AssocAndUpdate(Map, Measurement)
2:      IC_Map ← GetFirstCell(Map)
3:      for all IC_Meas in Measurement do
4:          if BorderIsAssociable(Map, IC_Meas) then
5:              IC_Assoc ← AssociatedCell
6:              UpdateBorder(IC_Assoc, IC_Meas)
7:          else
8:              IC_Assoc ← CreateCell(Map, IC_Meas)
9:          end if
10:         while IC_Map <= IC_Assoc do
11:             UpdateCell(IC_Map, IC_Meas)
12:             IC_Map ← GetNextCell(IC_Map)
13:         end while
14:     end for
15: end procedure
```

Fig. 6. Cell association and update algorithm

The association of existing and measured borders in the $BorderIsAssociable()$-routine is based on their lateral distance and the similarity of the bordering occupancy probabilities. The following $UpdateBorder()$-method performs a standard Kalman Filter update step, including computing the Kalman Gain:

$$K^{(t)} = \frac{\bar{\sigma}^2\left(y_{end}\right)^{(t)}}{\bar{\sigma}^2\left(y_{end}\right)^{(t)} + \sigma_z^2} \tag{9}$$

$$\hat{y}_{end}^{(t)} = \hat{y}_{end}^{(t-1)} + K^{(t)} \cdot \left(z_{end}^{(t)} - \hat{y}_{end}^{(t-1)}\right) \tag{10}$$

$$\sigma^2\left(y_{end}\right)^{(t)} = \left(1 - K^{(t)}\right) \cdot \bar{\sigma}^2\left(y_{end}\right)^{(t)} \tag{11}$$

As already stated, an approximated value for σ_z^2 from the previous step can be used to quantify the smoothness of the detected transition and to improve the estimated position and variance.

The update of a cell's posterior occupancy probability in $UpdateCell()$ is calculated by using Bayes' rule:

$$P\left(o|z^{(1:t)}\right) = \frac{P\left(z^{(t)}|o\right) \cdot P\left(o|z^{(1:t-1)}\right)}{P\left(z^{(t)}|z^{(1:t-1)}\right)} \tag{12}$$

$$= \frac{P\left(o|z^{(t)}\right) \cdot P\left(z^{(t)}\right) \cdot P\left(o|z^{(1:t-1)}\right)}{P\left(o\right) \cdot P\left(z^{(t)}|z^{(1:t-1)}\right)} \tag{13}$$

This equation is usually simplified by using the well-known log-odds formulation [2]. In doing so, logarithmized probability ratios are stored in the map and the posterior probabilities have to be recovered if needed. In our implementation, a subsequent merge algorithm takes place, which is based on comparing the occupancy probability values of neighboring cells. Our tests showed that the computational effort of

978-1-4799-2914-613/$31.00 ©2013 IEEE

recovering the probability posteriors during the merge step is higher than the benefit of using log-odds formulation for the probability update. For this reason, we reformulate equation 13 under the assumption $P(o) = 0.5$ as follows:

$$P(o|z^{(1:t)}) = P(o|z^{(t)})P(o|z^{(1:t-1)}) \; / \; \Big((1 - P(o|z^{(t)})) \cdot (1 - P(o|z^{(1:t-1)})) + P(o|z^{(t)} \cdot P(o|z^{(1:t-1)})\Big) \quad (14)$$

E. Merge Step

The computational effort of processing the previously presented steps strongly depends of the number of stored interval cells. To reduce the overall number, neighboring cells with similar probability posteriors can be combined in a subsequent merge step. In order to enable the creation of new cells with conflicting information, also the age $n_{age}^{(t)}$ of the investigated cells has to be considered. Both parameters, the minimum age and the maximum probability difference can be used to conveniently regulate the memory requirements of the resulting data structure.

V. Interaction with Object Tracking

For an application in ADAS, a map based environment representation of free and occupied areas has to be able to correctly deal with dynamic objects. In section II, we introduced the model based object module, which is able to provide information about dynamic objects in the vehicle's environment. The results of both modules can be combined according to the concept that has already been described in [1], [13]. The basic principles of this strategy can straightforwardly be adapted to the 2D Interval Occupancy Map. The different steps of the interaction are shown in figure 7 and explained in the corresponding following sections.

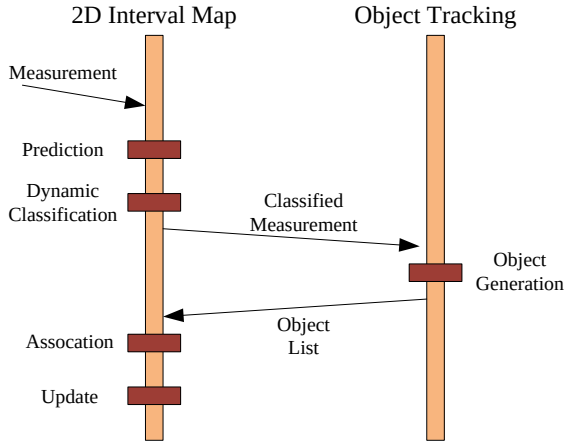

Fig. 7. Interaction between 2D Interval Map and object module

A. Prediction of dynamic cells

By using the introduced attribute $ID_{object}^{(t)}$, an interval cell can be associated with a dynamic object. In order to represent a moving object at the correct position in the occupancy map, these groups of dynamic cells have to be predicted before the map is updated. Therefore, an absolute lateral and longitudinal velocity of each dynamic cell is computed, depending on the velocities, accelerations and the yaw rate of the associated object. Based on this information the lateral and longitudinal shift of the cell is calculated and added to the cell's attributes. Similar to the ego compensation mechanism, cells are transfered to the neighboring interval, if the $x^{(t)}$-attribute exceeds the longitudinal border.

In our implementation, this task is incorporated in the prediction step introduced in the previous chapter. After the complete prediction step, the 2D Interval Map $\bar{M}^{(t)}$ represents the expected environment at the current measurement time stamp.

B. Dynamic classification of laser measurements

The idea of classifying the dynamic state of laser range measurements by comparing new measurements with previously accumulated maps has been presented several times [11], [1], [14]. This approach can straightforwardly be adapted to the 2D Interval Occupancy Map. Taking the longitudinal uncertainty and the lateral divergence of a single laser measurement into account, a rectangle can be constructed and compared to the predicted map $\bar{M}^{(t)}$. All measurements that hit a dynamic cell will be marked as $dynamic$, all other measurements will be classified according to the map's occupancy values. In case the measurement overlaps several cells with different probabilities, the maximum value will be processed. If the resulting probability goes below a threshold t_{free}, the measurement will be classified as $dynamic$, if it exceeds another threshold $t_{occupied}$, it will be categorized as $static$. In all other cases, the measurement will be classified as $unknown$.

In our system architecture this classification is performed by the map module in order to improve the successive object generation of the object module.

C. Dynamic Object association of map update

After the classification step, the object module estimates a new set of objects with corresponding dynamic states. In the map, this information is used to associate laser range measurements to dynamic objects in the feature extraction step. This association is based on a straightforward comparison of measurement and estimated object rectangle. If an extracted measurement is assigned to an object, a unique ID of the object will be stored in the cell's attribute $ID_{object}^{(t)}$ and will also be set in the selected interval cells during the association and update step. Please note that interval cells with conflicting object associations will not be merged in any processing step.

VI. Experimental Results

A. Accuracy

We evaluated the 2D Interval Occupancy Map by comparing the obtained results to our occupancy grid implementation and an additional ground truth. Our grid map implementation is based on a local, ego centered data structure. In order to provide a preview area of at least $70m$ at any

978-1-4799-2914-613/$31.00 ©2013 IEEE

time, the length and width of the grid was set to $140m$, at a cell size of $20cm$. Due to its constant alignment, a 2D Interval Map of $70m$ length is already able to provide a preview area of the required size at any time. Therefore, we use a 2D Interval Map of $70m$ length and $30m$ width in the following scenarios. The size of the intervals was set to $1m$, which turned out to be sufficient for the longitudinal accuracy in our tests.

All presented results were obtained by using an automotive laser scanner and a low cost inertial measurement unit (IMU) for the vehicle motion estimation. In order to obtain a ground truth of the obstacle's position in the vehicle's environment, we use a reference system, which is composed of a highly accurate IMU and DGPS.

Figure 8 shows the results from a static test scenario for a collision avoidance function. In this testbed, we measured the exact positions of the surrounding vehicles, pylons and guard rails by using the described reference system. These ground truth positions are depicted by the red lines and boxes in both representations in figure 8. The visualization of the resulting border estimation variances in the 2D Interval Map has been omitted for clarity. The comparison of grid and interval based

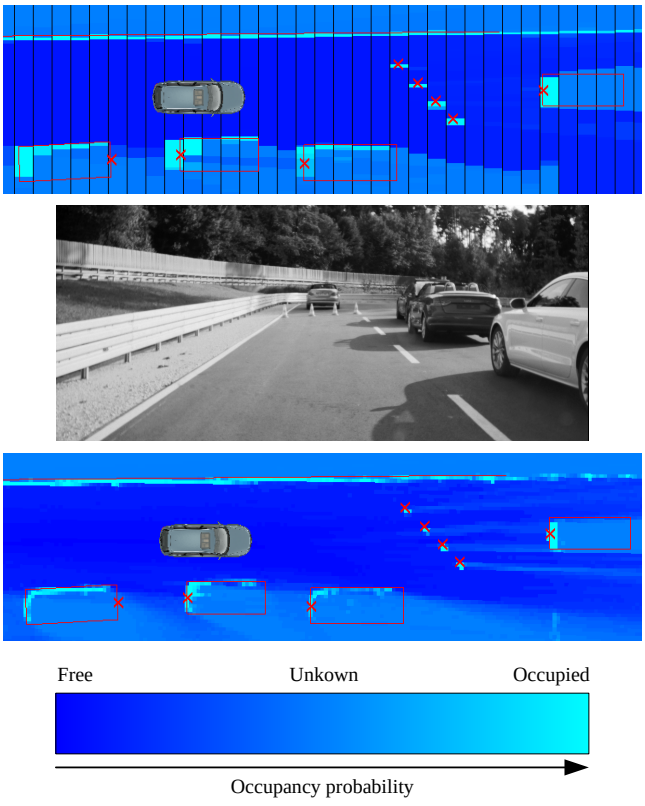

Fig. 8. Interval and grid based representation of collision avoidance scenario. Red lines and boxes indicate ground truth.

environment representation in this scenario illustrates the advantages and disadvantages of both approaches. On the one hand, longitudinal distances are, of course, better represented in the grid map. On the other hand, the lateral positions in the 2D Interval Map are not subject to discretization errors and therefore in some cases closer to the ground truth. Overall, the 2D Interval Map represents the environment in sufficient precision for typical state-of-the-art ADAS functions.

To confirm this fact, we also compared the results of a precrash ADAS function using different environment representations. In order to operate the collision avoidance system presented in [15], the relevant information of both environment representations has to be extracted, according to our concept introduced in section II. Based on this extracted information, the function analyzes different evasion trajectories, as depicted in figure 9 and calculates a potential crash distance. The distance values resulting from interval

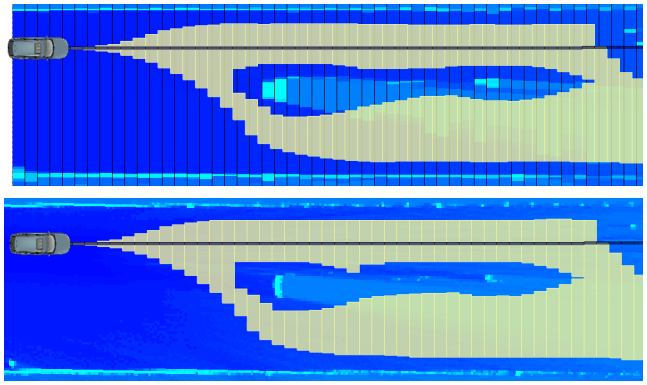

Fig. 9. Possible collision avoidance trajectories based on the 2D Interval Map and grid map

and grid based environment representation are compared in figure 10. A value of 0 indicates, that any crash can be avoided. Besides small deviations, the estimated distances and hence the behavior of the function are similar for both maps.

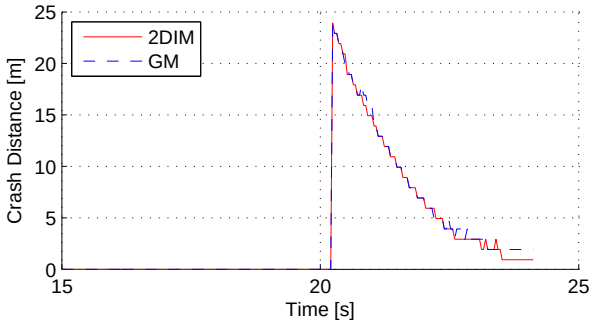

Fig. 10. Estimated crash distances based on the 2D Interval Map and grid map

B. Computational and Memory Requirements

For analyzing the computational effort and memory requirements of the 2D Interval Occupancy Map, we also analyzed an additional urban respectively highway scenario of approximately 1 minute duration. In all test data sets, we compared the effort for updating the map and extracting the collision avoidance information as explained above in both representations. The resulting mean computation times

978-1-4799-2914-613/$31.00 ©2013 IEEE

and their standard deviations in milliseconds are presented in table I. The results were obtained by using a standard laptop with Intel Core i7 m640 CPU. Regarding the average time for

Scenario	Collision avoidance		Urban		Highway	
	Mean	Std	Mean	Std	Mean	Std
2DIM Update	1.1	0.2	1.3	0.3	1.6	0.3
2DIM Update + Extraction	1.6	0.3	1.8	0.3	2.1	0.3
GM Update	5.1	0.8	5.8	1.0	5.5	0.9
GM Update + Extraction	7.7	1.2	8.3	1.4	8.1	1.2

TABLE I

MEAN COMPUTATION TIMES AND STANDARD DEVIATIONS IN MILLISECONDS OF THE 2D INTERVAL MAP AND THE GRID MAP IN DIFFERENT SCENARIOS

the complete update of the map, a substantial improvement of about 70% can be observed in all scenarios. The complete processing time for calculating the input parameters of the precrash function is even more reduced, as the extraction process also simplifies significantly.

Table II shows a comparison of the memory consumption of both representations. In this comparison, we only considered the storage of the map data structure, which means the interval cells respectively the grid cells. In contrast to the grid map's static size of $2.94MByte$, the 2D Interval Map has a scenario dependent memory demand, which is in average about 95% smaller.

Scenario		Collision avoidance	Urban	Highway
2DIM	Min	9.3 kB	13.0 kB	38.0 kB
	Max	82.4 kB	108.7 kB	108.9 kB
	Avg	45.6 kB	62.5 kB	79.3 kB
GM		2.9 MB	2.9 MB	2.9 MB

TABLE II

MEMORY CONSUMPTION OF THE 2D INTERVAL MAP AND THE GRID MAP

VII. CONCLUSIONS AND FUTURE WORK

A. Conclusions

In this paper, we presented a new way to describe occupancy information for ADAS functions. We showed, how posterior occupancy probabilities can be represented in interval cells of our 2D Interval Map framework. In the second part, we combined the 2D Interval Occupancy Map with a model based object tracking in order to correctly deal with dynamic objects in the vehicle's environment.

The presented experimental results show the great potential of the 2D Interval Map to represent occupancy information at low computation times and memory requirements. A tested collision avoidance function showed similar results for grid and interval based environment representation.

B. Future Work

A major task in future work is to further analyze the resulting occupancy information of the 2D Interval Map. The goal is to systematically compare the obtained results of an interval and grid based environment representation to a ground truth. Moreover, the impacts of discretization errors on ADAS functions in certain scenarios have to be studied.

Considering the memory requirements of the 2D Interval Occupancy Map, it would be interesting to define an upper memory limit. In this case, the merge step has to implement a policy which guarantees that the total number of existing cells does not exeed a predefined limit.

REFERENCES

[1] M. E. Bouzouraa, "Belegungskartenbasierte Umfeldwahrnehmung in Kombination mit objektbasierten Ansaetzen fuer fahrerassistenzsysteme," Dissertation, Technische Universitaet Muenchen, Muenchen, 2012.

[2] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. The MIT Press, Aug. 2005.

[3] T. Weiherer, E. Bouzouraa, and U. Hofmann, "A generic map based environment representation for driver assistance systems applied to detect convoy tracks," in *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, sept. 2012, pp. 691 – 696.

[4] I. Dryanovski, W. Morris, and J. Xiao, "Multi-volume occupancy grids: An efficient probabilistic 3d mapping model for micro aerial vehicles," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, oct. 2010, pp. 1553 –1559.

[5] M. Konrad, M. Szczot, F. Schule, and K. Dietmayer, "Generic grid mapping for road course estimation," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, june 2011, pp. 851 –856.

[6] M. Schmid, M. Maehlisch, J. Dickmann, and H.-J. Wuensche, "Dynamic level of detail 3d occupancy grids for automotive use," in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, june 2010, pp. 269 –274.

[7] J. Knaup and K. Homeier, "Roadgraph - graph based environmental modelling and function independent situation analysis for driver assistance systems," in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, sept. 2010, pp. 428 –432.

[8] 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 *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, Sept., pp. 514–521.

[9] R. Grewe, A. Hohm, S. Hegemann, S. Lueke, and H. Winner, "Towards a generic and efficient environment model for adas," in *Intelligent Vehicles Symposium (IV), 2012 IEEE*, June, pp. 316–321.

[10] Q. Baig, O. Aycard, T. D. Vu, and T. Fraichard, "Fusion between laser and stereo vision data for moving objects tracking in intersection like scenario," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, june 2011, pp. 362 –367.

[11] T.-D. Vu, O. Aycard, and N. Appenrodt, "Online localization and mapping with moving object tracking in dynamic outdoor environments," in *Intelligent Vehicles Symposium, 2007 IEEE*, june 2007, pp. 190 –195.

[12] C. Wang, C. Thorpe, S. Thrun, M. Hebert, and H. Durrant-Whyte, "Simultaneous localization, mapping and moving object tracking," *The International Journal of Robotics Research*, vol. 26, no. 9, p. 889916, 2007.

[13] K. Schueler, T. Weiherer, E. Bouzouraa, and U. Hofmann, "360 degree multi sensor fusion for static and dynamic obstacles," in *Intelligent Vehicles Symposium (IV), 2012 IEEE*, june 2012, pp. 692 –697.

[14] R. Matthaei, H. Dyckmanns, B. Lichte, and M. Maurer, "Motion classification for cross traffic in urban environments using laser and radar," in *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on*, july 2011, pp. 1 –8.

[15] M. Reichel, M. Bouzouraa, A. Siegel, K. Siedersberger, and M. Maurer, "Erweiterte Umfelderkennung und Nutzung einer Ausweichanalyse als Grundlage einer aktiven Gefahrenbremsung," in *Automatisierungssysteme, Assistenzsysteme und eingebettete Systeme für Transportmittel (AAET)*, Braunschweig, 2010.