2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

Extending occupancy grid mapping for dynamic environments

Joseph Wessner[1], Wolfgang Utschick[2]

Abstract— **In this paper, the commonly used filtering technique *occupancy grid mapping* for static environments is extended for dynamic environments. The proposed method is able to estimate velocities indirectly. We apply a distribution model of the respective state variable to estimate the cell dynamics by means of prediction and update cycle, as known by standard tracking filters. Therefore, we present a straight forward derivation of the prediction and update rule. Furthermore, we validate our approach by simple one dimensional simulations, and show how it can be extended into a two dimensional world, including the resulting consequences, e.g. in terms of memory requirements.**

I. Introduction

The development for autonomous driving presents new challenges for the environment perception. An autonomous vehicle needs a precise representation of its surrounding environment to be able to act accordingly. Even today's advanced driver assistance systems (ADAS) require detailed information about the obstacles around themselves. Besides knowing where other vehicles are located, it is also necessary to know where no obstacle is located or in other words to know about free space. Only with the knowledge of free space a safe trajectory can be planned and executed without creating dangerous situations.
To be able to provide this information in the required quality, today's ADAS rely on multiple sensors instead of a single one (e.g. radar, lidar, camera, ...). Since only a fusion of different sensor data can reliably provide the required environment representation. The fusion of different sensor data not only provides more accurate data, but can also handle drawbacks or failures of single sensors.
One very promising technique for sensor data fusion, is the occupancy grid mapping originally introduced by Moravec and Elfes in 1985 [1]. This technique is the most common one for static environments and can directly be used for the fusion of different sensors. Using efficient storage techniques as described in [2] the grid mapping framework even allows the mapping of huge static environments. This form of mapping even allows very efficient and generic SLAM approaches, as described in [3].
To use this approach also for dynamic environments as in the automotive field, various approaches have been developed. One approach for taking advantage of the occupancy grid mapping for sensor fusion and being able to handle the dynamic environments in the automotive field is described in [4] and [5]. Both publications combine the data of radar sensors and laser scanners using the occupancy grid mapping. The resulting occupancy grid map is then used to classify cells as being dynamic or static, the dynamic cells are then used as measurement input for object tracking. In [6], a grid map using the Dempster Shafer theory acts as only sensor for the object tracking, where the conflicting probabilities of free and occupied act as dynamic belief.
In [7], particles are used to model the occupancy and velocity distribution for each independent cell. The measured occupancy of cells is used as weighting factor for the particles. The prediction of the particles is compensated by the own ego motion and based on a simple linear motion model. Due to the nature of the filtering algorithm this approach is able to estimate motion within the map cells.
In this paper, we adopt the idea to model a velocity distribution for each cell as published in [8] and [9]. Our contribution is to provide a straight forward derivation of the prediction and update rule for the dynamic occupancy grid. We validate our approach with two one dimensional scenarios and discuss practical limits for two dimensional grids.
The paper is organized as follows, Section II introduces some definitions and notations and resumes a short derivation of the update formula in the static case. In Section III, we extend the static occupancy grid mapping with an update rule to handle dynamic environments. Section IV shows experimental results of the dynamic occupancy grid in a one dimensional world, whereas Section V presents the extension to a two dimensional world as well as some resulting limitations. Section VI gives a short summary and some ideas for further research.

II. Static occupancy grid

A. Definitions

We use following variables and notation in this paper:

- $c \in \mathcal{C} \subseteq \mathbb{Z}^n$ defines the coordinate of a grid cell, and corresponds to the respective dimension of the grid map accordingly (e.g., either 1D, 2D or 3D), the coordinate indication is used as subscript text, e.g. $\boxdot_c$
- $t \in \mathcal{T} \subseteq \mathbb{N}$ describes a discrete time stamp, it is used as superscript text, e.g. $\boxdot^t$. A series of time stamps is abbreviated by the first and last time stamp using a colon, e.g., time from time stamp 1 till time stamp 7 is written as $\boxdot^{1:7}$.
- $O \in \mathcal{O} = \{\text{occ}, \text{free}\}$ represents the binary random variable of the state being either occupied (occ) or free (free). Note: $P(O = \text{occ}) = 1 - P(O = \text{free})$.

[1]Joseph Wessner is with Zukunft Mobility GmbH, Kösching, Germany joseph.wessner.zkm@zf.com
[2]Wolfgang Utschick is with the Department of Electrical and Computer Engineering, Technische Universität München, Germany utschick@tum.de

- $Z \in \mathcal{Z}$ defines the random variable of an arbitrary sensor measurement. Z^t denotes the sensor measurement at time stamp t.
- $V \in \mathcal{V} \subseteq \mathbb{Z}^n$ represents the random variable of velocities. The velocity is given as cells per time stamp. The dimension of the velocities corresponds to the dimension denoted by $\mathcal{C}$.

B. Updating

In static environments the occupancy of a single cell is not changing over time, except for incorporation of new knowledge given new sensor measurements, as there is no dynamic component by definition. Therefore, we have to store only the probability of being occupied for every cell.
The assumption of a static environment is the most common use-case for occupancy grid maps. For computational reasons, we furthermore assume that each grid cell is independent of its neighbor cells.
The goal of the occupancy grid mapping is to give a probability of how likely a cell is being occupied (or free) after a given set of measurements: [1]

$$P(O_c|Z^{1:t}). \tag{1}$$

Using Bayes rule $P(O_c|Z^{1:t})$ can be calculated as:

$$\begin{aligned} P(O_c|Z^{1:t}) &= P(O_c|Z^t, Z^{1:t-1}) \\ &= \frac{P(Z^t|O_c, Z^{1:t-1})\ P(O_c|Z^{1:t-1})}{P(Z^t|Z^{1:t-1})}. \end{aligned} \tag{2}$$

We furthermore assume that O_c contains all relevant information of the measurements, which have been used for updating so far. Therefore we can substitute $P(Z^t|O_c, Z^{1:t-1})$ to $P(Z^t|O_c)$ leading to following simplification of (2):

$$P(O_c|Z^{1:t}) = \frac{P(Z^t|O_c)\ P(O_c|Z^{1:t-1})}{P(Z^t|Z^{1:t-1})}. \tag{3}$$

Applying Bayes rule again, we obtain:

$$P(O_c|Z^{1:t}) = \frac{P(O_c|Z^t)\ P(Z^t)\ P(O_c|Z^{1:t-1})}{P(O_c)\ P(Z^t|Z^{1:t-1})}. \tag{4}$$

As we use a binary random variable O, we can use the odds-ratio instead of the probability values. The probability value $P(X)$ of a binary random variable can be easily converted to the odds-ratio $R(X)$ and back:

$$\frac{P(X)}{1-P(X)} = R(X)$$
$$P(X) = \frac{R(X)}{1+R(X)}$$

Using the odds-ratio simplifies the above expressions by removing factors, which are hard to compute (as $P(Z^t|Z^{1:t-1})$).

With the odds-ratio the formula for estimating the occupancy for a single cell after various measurements (4) becomes:

$$\begin{aligned} &\frac{P(O_c|Z^{1:t})}{1-P(O_c|Z^{1:t})} = \\ &\frac{P(O_c|Z^t)}{1-P(O_c|Z^t)} \cdot \frac{P(O_c|Z^{1:t-1})}{1-P(O_c|Z^{1:t-1})} \cdot \frac{1-P(O_c)}{P(O_c)}. \end{aligned} \tag{5}$$

The probability of cell c being occupied after t sensor measurements ($P(O_c|Z^{1:t})$) only depends on:

- an update term $P(O_c|Z^t)$, which denotes the conditional probability of being occupied given only the current measurement Z^t,
- the previous estimation $P(O_c|Z^{1:t-1})$, after all previous measurements apart the latest one (Z^t), and
- a prior probability of being occupied given no measurement data $P(O_c)$.

Most common the prior occupancy probability $P(O_c)$ is set to 50%, meaning that the probability of being occupied equals the probability of being free. This additional assumption eliminates the third factor in equation 5 and makes the update formula even more handy to use:

$$\frac{P(O_c|Z^{1:t})}{1-P(O_c|Z^{1:t})} = \frac{P(O_c|Z^t)}{1-P(O_c|Z^t)} \cdot \frac{P(O_c|Z^{1:t-1})}{1-P(O_c|Z^{1:t-1})}. \tag{6}$$

This final update rule[2] (6) allows the implementation of a recursive algorithm for estimating the occupancy probability in static environments.

III. Dynamic occupancy grid

A. Extending for dynamics

In dynamic environments, the occupancy of grid cells change over time as objects occupying certain cells are moving and therefore occupy different cells in future time stamps. To address this (possible) change over time, we have to modify the update formula (5). Rather than updating conditional probabilities of time-invariant occupancies (O_c), we have to consider different time stamps for the occupancy probability of the cells:

$$\begin{aligned} &\frac{P(O_c^t|Z^{1:t})}{1-P(O_c^t|Z^{1:t})} = \\ &\frac{P(O_c^t|Z^t)}{1-P(O_c^t|Z^t)} \cdot \frac{P(O_c^t|Z^{1:t-1})}{1-P(O_c^t|Z^{1:t-1})} \cdot \frac{1-P(O_c^t)}{P(O_c^t)}. \end{aligned}$$

To model the lack of knowledge about the occupancy of a cell without any measurement data, we also assume $P(O_c^t = \text{occ}) = P(O_c^t = \text{free}) = 0.5$, which simplifies the above equation to (similar to Eq. (6)):

$$\frac{P(O_c^t|Z^{1:t})}{1-P(O_c^t|Z^{1:t})} = \frac{P(O_c^t|Z^t)}{1-P(O_c^t|Z^t)} \cdot \frac{P(O_c^t|Z^{1:t-1})}{1-P(O_c^t|Z^{1:t-1})}. \tag{7}$$

[1]The following derivation of the update formulas for the static environment are based on [10, Chapters 4.2,9.1,9.2] and [11].

[2]Most implementations use the logarithmic variant of this formula for performance reasons. As this does not affect the math behind it, we skip the logarithm here.

The conditional probability $P(O_c^t|Z^{1:t})$ of a given cell c to be occupied at time stamp t depending on all measurement data up to time t can be calculated with:

- the same inverse sensor model used in Eq. (5), which maps the current measurement data Z^t to the current grid cell's occupancy probability $P(O_c^t|Z^t)$, and
- a prediction function, which predicts the occupancy probability for every grid cell for time stamp t depending only on the previous measurements ($Z^{1:t-1}$): $P(O_c^t|Z^{1:t-1})$.

B. Prediction formula

For the prediction function $P(O_c^t|Z^{1:t-1})$, we make following assumptions:

- the velocity of movements is constant and the motion follows a linear model,
- given a specific velocity, each cell is reachable exactly from one other cell (or the same cell if $v = 0$). The coordinates of that source cell can be calculated using: $c_{\text{source}} = c_{\text{destination}} - v$,
- only occupancy and not freeness has a velocity,
- the initial velocity distribution (as well as the initial occupancy distribution) for every cell is uniform, and
- cells outside the mapped area are treated as unknown cells with the initial probability distributions.

Following the above assumptions, the prediction of the occupancy grid map can be divided into the following three steps. First of all, the joint distribution of being occupied with a given velocity is calculated:

$$\begin{aligned}
&P(O_c^t = \text{occ}, V_c^t = v|Z^{1:t-1}) = \\
&P(O_{c-v}^{t-1} = \text{occ}, V_{c-v}^{t-1} = v|Z^{1:t-1}) = \\
&P(O_{c-v}^{t-1} = \text{occ}|Z^{1:t-1}) \cdot P(V_{c-v}^{t-1} = v|O_{c-v}^{t-1} = \text{occ}, Z^{1:t-1}).
\end{aligned} \tag{8}$$

The prediction of the joint distribution $P(O_c^t = \text{occ}, V_c^t = v|Z^{1:t-1})$ only depends on the occupancy probability and the conditional velocity probability of the ancestor cell. The coordinates of the ancestor cell are the coordinates of the current cell minus the velocity, as we store the velocities as cells per time stamp.

After the calculation of the joint distribution over all possible velocity values, the occupancy probability can be calculated by marginalization:

$$\begin{aligned}
&P(O_c^t = \text{occ}|Z^{1:t-1}) = \\
&\quad \sum_{v \in V} P(O_c^t = \text{occ}, V_c^t = v|Z^{1:t-1}).
\end{aligned} \tag{9}$$

This predicted occupancy probability is used in Eq. (7) to get the dynamic occupancy probability for time stamp t.

The third calculation step within the prediction is to update the velocity distribution for the given cell. As we already know the joint probability distribution of occupancy and velocity (8) and the occupancy probability (9), the probability distribution of velocities depending on the occupancy can be computed with following formula:

$$\begin{aligned}
&P(V_c^t = v|O_c^t = \text{occ}, Z^{1:t-1}) = \\
&\frac{P(O_c^t = \text{occ}, V_c^t = v|Z^{1:t-1})}{P(O_c^t = \text{occ}|Z^{1:t-1})}.
\end{aligned} \tag{10}$$

Note, that this third computation (10) is important for the prediction in the next time step, as it is needed for the joint probability calculation (8). The initial value of the velocity distribution is uniform, meaning each possible velocity is equally likely:

$$P(V_c^0 = v|O_c^0 = \text{occ}) = \frac{1}{|\mathcal{V}|}.$$

C. Forget old measurements

We furthermore extend the prediction formula (8) by a forgetting factor ε, which represents the weight of the new state being equal to the initial state:

$$\begin{aligned}
&P(O_c^t = \text{occ}, V_c^t = v|Z^{1:t-1}) = \\
&\quad \boldsymbol{(1-\varepsilon)} \cdot P(O_{c-v}^{t-1} = \text{occ}|Z^{1:t-1}) \\
&\qquad \cdot P(V_{c-v}^{t-1} = v|O_{c-v}^{t-1} = \text{occ}, Z^{1:t-1}) \\
&\quad + \boldsymbol{\varepsilon} \cdot P(O_c^0 = \text{occ}) \cdot P(V_c^0 = v|O_c^0 = \text{occ}).
\end{aligned} \tag{11}$$

As the initial values for $P(O_c^0 = \text{occ})$ and $P(V_c^0 = v|O_c^0 = \text{occ})$ are the same for every $c \in \mathcal{C}$ and $v \in \mathcal{V}$, we can substitute the last term in the above equation with the initial values:

$$\varepsilon \cdot P(O_c^0 = \text{occ}) \cdot P(V_c^0 = v|O_c^0 = \text{occ}) = \frac{\varepsilon}{2 \cdot |\mathcal{V}|}.$$

With introducing the forgetting factor ε, it is obvious that the dynamic occupancy map will approach to its initial state if no updates during measurements can be applied.
Furthermore, the forgetting factor prevents the probability values from getting too close to 1 or 0, which would lead to numeric instabilities.

D. Static map as special dynamic case

The static occupancy grid is only a special case of the dynamic occupancy grid. In the static use-case, the cell's probability of having a velocity of 0 is 100%, therefore $P(V_c^t = 0|O_c^t = \text{occ}, Z^{1:t-1}) = 1$ for all possible values of t, whereas $P(V_c^t = v|O_c^t = \text{occ}, Z^{1:t-1}) = 0$ for all velocity values $v \neq 0$. This leads to following simplification of the occupancy probability calculation used during the prediction step (9):

$$\begin{aligned}
P(O_c^t = \text{occ}|Z^{1:t-1}) &= P(O_c^t = \text{occ}, V_c^t = 0|Z^{1:t-1}) \\
&= P(O_c^{t-1} = \text{occ}|Z^{1:t-1}).
\end{aligned}$$

Inserting this identity "prediction" into the occupancy update Eq. (7), directly leads to the update formula used in the static occupancy grid (5).

IV. Experiments in 1D

A. Setup

For our experiments in the one-dimensional case we use:

- a grid map with 151 cells ($\mathcal{C} = \{c \in \mathbb{Z} : 0 \leq c \leq 150\}$),
- a distance measuring sensor located at $c = 0$,
- a fixed set of allowed velocities ($\mathcal{V} = \{v \in \mathbb{Z} : -3 \leq v \leq +3\}$), and
- a total timespan of 25 ($\mathcal{T} = \{t \in \mathbb{N} : t \leq 25\}$).

To obtain the occupancy based on the sensor measurements, we use a very basic inverse sensor model, which assumes free space up to the measured distance, occupied cells at the measured distance and gives no information beyond the measured distance. To account for measurement noise, we model the uncertainty with a quadtric polynomial around the measured distance value. The mapping of the cell occupancy (O_c) given the measurement (Z) is given in the following equation and shown in Figure 1:

$$f_{a,b,d}(x) = \begin{cases} a & , 0 < x < d - \alpha \\ \frac{a-b}{\alpha^2}(x-d)^2 + b & , d - \alpha \leq x < d \\ \frac{0.5-b}{\alpha^2}(x-d)^2 + b & , d \leq x < d + \alpha \\ 0.5 & , d + \alpha \leq x \end{cases} \tag{12}$$

The function for the inverse sensor model relies on:

- the measured distance d,
- a value a defining the occupancy probability for cells in front of the actual measurement,
- a value b defining the occupancy probability for the measured distance itself,
- and a constant α defining the range of cells considered to be occupied.

In our experiments, we use a conservative value $a = 0.4$ for updating "free" cells, $b = 0.8$ for updating the actual measured cell, and, a forgetting factor of $\varepsilon = 0.08$.

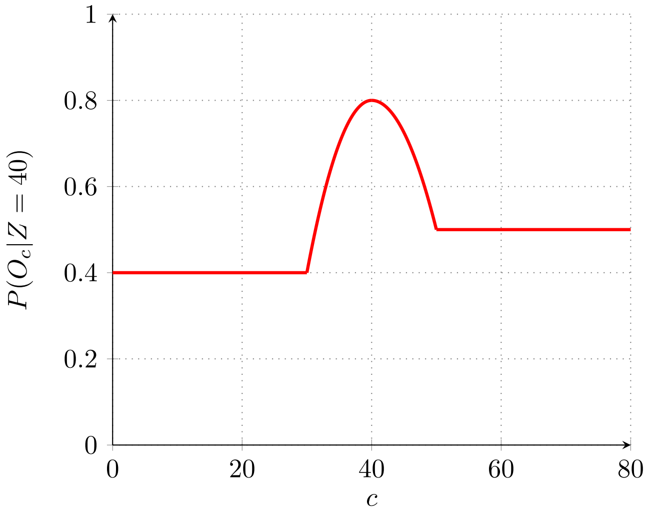
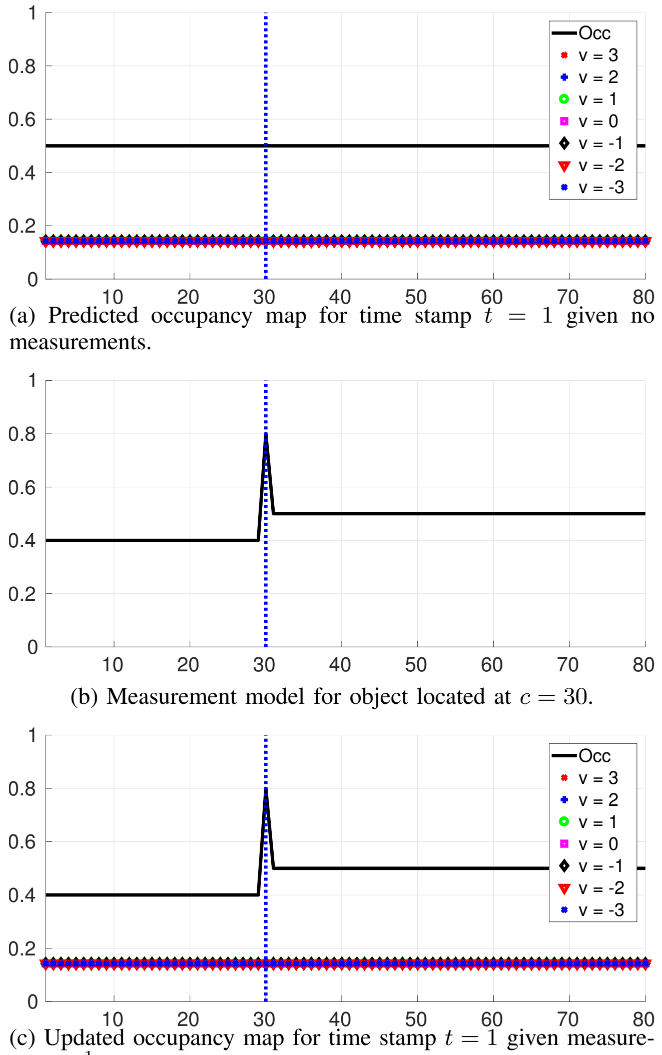


Fig. 1: Inverse sensor model $P(O_c|Z = d)$ with $d = 40$, $a = 0.4$, $b = 0.8$, and $\alpha = 10$ (Eq. 12).

B. Simple linear noise-free movement

In the first experiment, the target moves with a constant velocity ($v = 2$), starting at cell $c = 30$. The measurement of the distance is noise-free and therefore we choose $\alpha = 1$.



(a) Predicted occupancy map for time stamp $t = 1$ given no measurements.

(b) Measurement model for object located at $c = 30$.

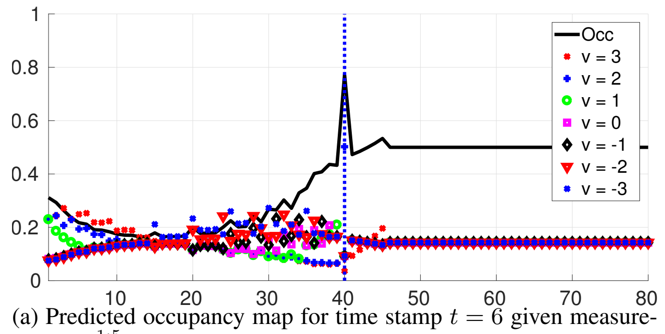
(c) Updated occupancy map for time stamp $t = 1$ given measurement Z^1.

Fig. 2: Predicted occupancy map, measurement model, and updated occupancy map for $t = 1$. The blue vertical line denotes the real position (ground-truth).

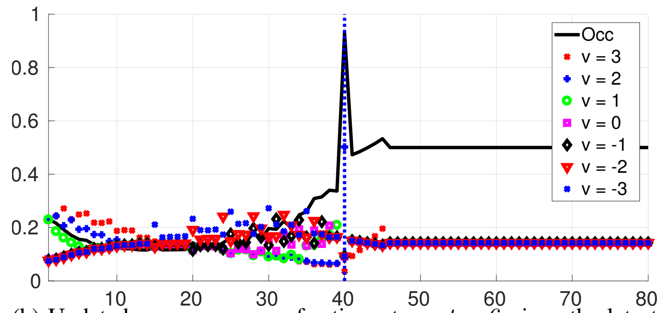
Figure 2 shows the predicted map, the measurement, and the updated map at time stamp $t = 1$. The prediction for $t = 1$ (Figure 2a) is not based on any measurement data and therefore it is equivalent to the initial value of being occupied: $P(O_c^1|Z^{1:0}) = P(O_c^0) = \frac{1}{2}$. The same argumentation holds true for the velocity distribution $P(V_c^0 = v|O_c^0) = \frac{1}{|V|} = \frac{1}{7}$, which is also shown in Figure 2a.

The measurement model (Figure 2b) shows a lower occupancy probability ($P(O = \text{occ}) = 0.4$) up to the object's location at $c = 30$, a high occupancy probability ($P(O) = 0.8$) at the object's location, and basically no information behind the object ($P(O) = 0.5$).

Due to the update in Eq. (7), the updated occupancy map (Figure 2c) incorporates the measurement data for the occupancy probability, while the velocity distribution stays untouched.



(a) Predicted occupancy map for time stamp $t = 6$ given measurements $Z^{1:5}$.



(b) Updated occupancy map for time stamp $t = 6$ given the latest measurement Z^6.

Fig. 3: Predicted occupancy map and updated occupancy map for $t = 6$. The blue vertical line denotes the real position (ground-truth).

Figure 3 illustrates the probability distribution after only a few (5) time stamps. As mentioned in the beginning this scenario is based on an object with a constant velocity of $v = +2$, therefore the object is now positioned at $c = 40$, which is also indicated by the vertical blue (ground-truth) line.

The predicted occupancy value for cell $c = 40$, based only on the previous measurements $P(O^6_{40} = \text{occ}|Z^{1:5}) = 0.77$ is high, whereas the predicted occupancy probability of all cells $c < 40$ is below 0.5 (cf. Figure 3a). Even the previous ($t = 5$) object location at $c = 38$ is now predicted to be more likely being free then occupied: $P(O^6_{38} = \text{occ}|Z^{1:5}) = 0.44$. More interesting than the predicted occupancy is the predicted velocity distribution, the most likely velocity for cell $c = 40$ is the ground truth velocity of $v = +2$ with a probability of $P(V^6_{40} = +2|O^6_{40} = \text{occ}, Z^{1:5}) = 0.50$. This small example shows, that the proposed system is able to estimate a cell velocity without explicit velocity measurements.

The measurement model $P(O^6_c|Z^6)$ is not shown in Figure 3 as it is basically the same as for time stamp 1 shown in Figure 2b, expect that the occupied location changed from $c = 30$ to $c = 40$.

Figure 3b illustrates the updated dynamic occupancy map $P(O^6_c|Z^{1:6})$ based on the predicted map $P(O^6_c|Z^{1:5})$ and the latest measurement information $P(O^6_c|Z^6)$. As the proposed measurement model only provides estimates of occupancy probabilities, the velocity distribution stays the same as in the predicted map.

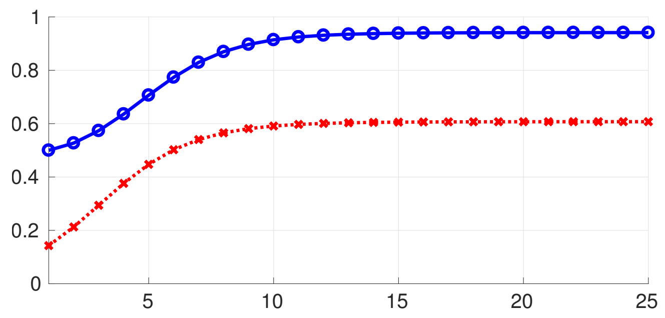


Fig. 4: Blue line shows the occupancy probability $P(O^t_c = \text{occ}|Z^{1:t})$ at the object's position over time, whereas the red line shows the corresponding velocity probability $P(V^t_c = +2|O^t_c = \text{occ}, Z^{1:t-1})$ over time.

Eventually, Figure (4) shows the occupancy probability $P(O^t_c|Z^{1:t})$ and the velocity probability of the object's speed $P(V^t_c = +2|O^t_c, Z^{1:t-1})$ at the object position over time. As already shown in the previous figures (cf. 2 and 3) the dynamic occupancy map reflects our scenario very quickly. This Figure also shows that the proposed approach is able to handle very simple scenarios.

C. Simple linear movement with noise

In the second example, we use the same constant moving target as in the previous scenario. However, we add a random Gaussian distributed error with a standard deviation of 2.5 (cells) to the distance measurement. To compensate for the noisy measurement, we choose $\alpha = 5$. Figure 5 shows a measurement at $t = 10$, where the error is 3 (cells) in distance.

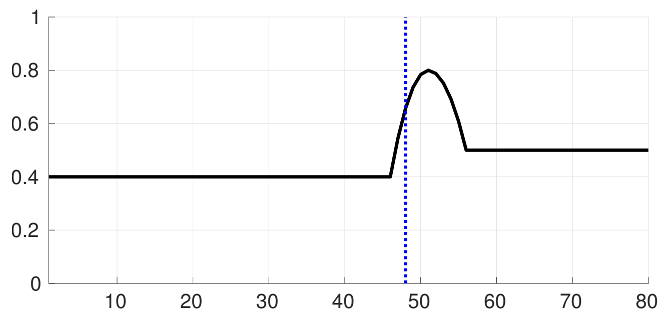


Fig. 5: Measurement model $P(O^{10}_c|Z^{10})$ with $\alpha = 5$. The error between the peak ($c = 51$) and the ground-truth location ($c = 48$, blue line) is clearly visible.

As expected the resulting dynamic occupancy map is not as accurate as in the previous scenario (cf. figure 6). Nevertheless, the occupancy probability at the ground-truth position is above 90% after only 7 prediction and measurement cycles ($P(O^7_{42} = \text{occ}|Z^{1:7}) = 0.9466$). More interesting than the occupancy probability is the velocity distribution, besides the ground-truth velocity of $v = +2$ being the most probable one, the two adjacent velocities $v \in \{+1, +3\}$ have the highest probability in the neighboring cells. The explanation for this behavior is as follows: Based on the previous object location, if the object is at a lower position compared to the ground-truth position ($c < c_{\text{ground-truth}}$), then it must have had a lower velocity ($v < +2$) and this matches with the stored velocity probability given that the cell is occupied: $P(V_c^t | O_c^t = \text{occ}, Z^{1:t-1})$. The same argumentation holds true for the other neighbor cells, cells with higher indices have moved faster, resulting in a higher predicted velocity value, whereas cells with lower indicies have a lower predicted velocity value.

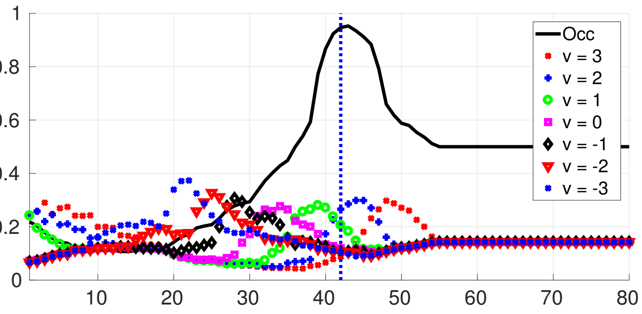


Fig. 6: Updated occupancy map ($P(O_c^7 | Z^{1:7})$ at time $t = 7$.

Figure 7 shows a steady growth in the map's belief of the correct velocity (red). Although the probability values of $P(V_c^t = +2 | O_c^t = \text{occ}, Z^{1:t-1})$ are much smaller than in the noise-free scenario, the most likely velocity at the object's location was, besides a small initial phase ($t < 6$), always the correct one. Both the ups and downs in the occupancy probability as well as the slower growing confidence in the right velocity (compared to the previous example) are due to the introduction of the measurement error. Nevertheless, this example shows, that the proposed approach can handle uncertainties due to noisy sensor measurements. Needless to say, that the resulting output of the dynamic occupancy grid suffers from input errors.

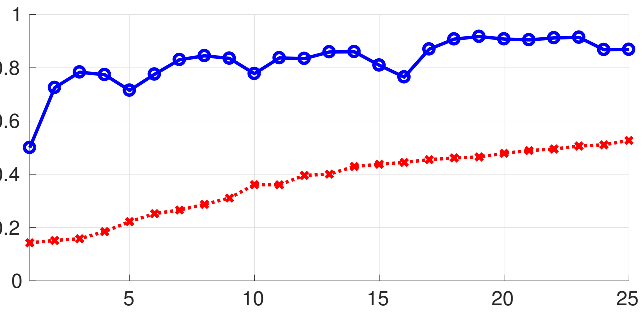


Fig. 7: Occupancy probability $P(O_c^t = \text{occ}|Z^{1:t})$ (blue) and velocity probability $P(V_c^t = +2|O_c^t = \text{occ}, Z^{1:t-1})$ (red) at the object's real position over time.

V. Extending to Second Dimension

A. Extension

The previous section showed that the proposed dynamic occupancy grid works well in an one dimensional case. Nevertheless, all formulas and also the derivation of them made no assumption about restricting the approach to only one dimension. Therefore, it can be expected that this approach can also handle the two dimensional case.

The two most obvious changes for a two dimensional dynamic map are:

- the coordinate c has now two components, one for each dimension: $c = (c_x, c_y)^\top \in \mathcal{C} \subseteq \mathbb{Z}^2$, and
- the velocity V is also expressed using two components: $v = (v_x, v_y)^\top \in \mathcal{V} \subseteq \mathbb{Z}^2$.

The set of possible values for the velocities $\mathcal{V}$ has to be defined using the absolute velocity value:

$$\mathcal{V}_{\text{2D}} = \{(v_x, v_y)^\top \in \mathbb{Z}^2 \mid \sqrt{v_x^2 + v_y^2} \le v_{\max}\}.$$

The same argumentation holds for extending our approach to an arbitrary number of dimension.

B. Memory consumption

The proposed method is principally not limited by any particular maximum set of coordinates or dimensions. On the other hand, some practical limits have to be taken into account when implementing the respective approach. One major limitation of this approach is the required memory. In our approach, we have to store two probability distributions per cell:

- the current occupancy probability $P(O_c^t = \text{occ}|Z^{1:t})$, and
- the velocity distribution $P(V_c^t = v|O_c^t = \text{occ}, Z^{1:t-1})$.

As occupancy values are binary, we only have to store one value for the occupancy probability. The velocity distribution on the other hand requires to store one value for every possible velocity, leading to $|\mathcal{V}|$ probability values. In total, we have to store exactly $1 + |\mathcal{V}|$ probability values per cell.

The total number of cells in the grid is defined by the dimension of the grid map and the resolution of the grid (normally expressed as cell size). As the velocities are expressed as “cells per time stamp” the number of possible velocity values depends on the frequency the grid update/predict cycles are computed.

As an example we take following setup from the automotive area:

- the grid should cover an area of $200\text{m} \times 80\text{m}$ around the vehicle ($\text{dim}_\text{x} = 200\text{m}, \text{dim}_\text{y} = 80\text{m}$),
- one cell should cover an area of $0.1\text{m} \times 0.1\text{m}$ leading to: $\text{cellSize} = 0.1\text{m}$,
- the system should do the predict/update cycle every 100ms or with a frequency $f = 10\text{Hz}$, and
- the maximum allowed velocity value should be $v_{\max} = 252\frac{\text{km}}{\text{h}} = 70\frac{\text{m}}{\text{s}}$.

The total number of required cells can be calculated straight forward:

$$\begin{aligned} N &= \frac{\mathrm{dim_x}}{\mathrm{cellSize}} \cdot \frac{\mathrm{dim_y}}{\mathrm{cellSize}} \\ &= \frac{200}{0.1} \cdot \frac{80}{0.1} \\ &= 2\ 000 \cdot 800 = 1\ 600\ 000. \end{aligned}$$

To get the number of possible velocities, $v_{\max}$ has to be converted to "cells per time stamp":

$$v_{\max} = \frac{70\frac{\mathrm{m}}{\mathrm{s}}}{10\mathrm{Hz} \cdot 0.1\mathrm{m}} = 70.$$

The total number of possible velocities $|\mathcal{V}|$ can be approximated using the formula for calculating the area of a disk:

$$|\mathcal{V}| \approx v_{\max}^2 \pi \approx 15\ 394.$$

As we have to store probability values, we choose the single precision *float* data type to store our values. The size of a *float* is 4B (byte). Therefore, we get a total memory requirement of:

$$\begin{aligned} \mathrm{MemoryRequirement} &= N \cdot (1 + |\mathcal{V}|) \cdot 4\mathrm{B} \\ &= 1\ 600\ 000 \cdot (1 + 15\ 394) \cdot 4\mathrm{B} \\ &= 98\ 528\ 000\ 000\mathrm{B} \\ &\approx 92\mathrm{GiB}. \end{aligned}$$

This huge memory requirement is practically not available on today's computers and even if enough memory would be available, processing this amount of data within the time constraint of 100ms is unfeasible. To reduce the amount of required memory, some sort of compression has to be applied. Approaches like [2] promises a huge saving in terms of memory consumption by reducing the number of cells. This, or similar approaches, should also increase the runtime performance of our method.

VI. Conclusion

In this paper, we presented a straightforward approach how to extend a static occupancy grid to a dynamic occupancy grid. Section III described the underlying theory and derivation of the update rules. In Section IV, we demonstrated that this approach works using two simple one dimensional examples. The previous section (Section V) made clear that our approach is principally able to handle the two dimensional case, but that it fails due to memory and processing capacity boundaries in reality.

The main focus in our further work will therefore lie in how to optimize this approach in terms of implementation to deal with these constrains without loosing too much information or flexibility. The easiest way to getting closer to practical memory and processing requirements is to increase the cell size as an increase in the cell size will result in a quartic decrease of memory requirement. The costs of increasing the cell size on the other hand will be a decrease in accuracy. Therefore, a better way of reducing the complexity is needed.

As some sensors are able to provide velocity information in their measurement (e.g. radar sensors), this information should be incorporated in the grid update step. The current formulation of our approach just uses occupancy information of the measurement data and estimates the velocity information only indirect. An enhanced update formulation taking measured velocities into account promises faster and better velocity estimates and therefore needs to be incorporated as well in the future.

Due to the nature of the update formulas (Eq. 8 - 10), each cell prediction can be calculated independently of each other with the information of the previous grid map. Therefore, this approach is predestinated for a parallel implementation using the computing power of modern GPUs.

References

[1] H. Moravec and A. Elfes. High resolution maps from wide angle sonar. *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, 2:116–12, 1985.

[2] A. Hornung, K.M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard. OctoMap: an efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 34(3):189–206, feb 2013.

[3] R. Ait-jellal and A. Zell. Outdoor Obstacle Avoidance based on Hybrid Visual Stereo SLAM for an Autonomous Quadrotor MAV. *European Conference on Mobile Robots (ECMR), 2017*, pages 1–8, 2017.

[4] R. Garcia, O. Aycard, T.D. Vu, and M. Ahrholdt. High level sensor data fusion for automotive applications using occupancy grids. *2008 10th International Conference on Control, Automation, Robotics and Vision*, pages 530–535, 2008.

[5] 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. *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 294–300, 2010.

[6] R. Jungnickel and F. Korf. Object tracking and dynamic estimation on evidential grids. *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 2310–2316, 2014.

[7] R. Danescu, F. Oniga, S. Nedevschi, and M.M. Meinecke. Tracking multiple objects using particle filters and digital elevation maps. *2009 IEEE Intelligent Vehicles Symposium*, pages 88–93, 2009.

[8] C. Chen, C. Tay, C. Laugier, and K. Mekhnacha. Dynamic environment modeling with gridmap: A multiple-object tracking application. *9th International Conference on Control, Automation, Robotics and Vision, 2006, ICARCV '06*, pages 1–6, 2006.

[9] T. Yuan, D. Nuss, and G. Krehl. Fundamental Properties of Dynamic Occupancy Grid Systems. *IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 153–156, 2015.

[10] S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics*. 2005.

[11] G.D. Tipaldi and W. Burgard. Robot Mapping - Grid Maps. http://ais.informatik.uni-freiburg.de/teaching/ws17/mapping/pdf/slam10-gridmaps.pdf, 2017. [Online; accessed 11-November-2017].