2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Recursive Importance Sampling for Efficient Grid-Based Occupancy Filtering in Dynamic Environments

Sebastian Brechtel, Tobias Gindele and Rüdiger Dillmann
Institute for Anthropomatics
Karlsruhe Institute of Technology (KIT)
D-76128 Karlsruhe, Germany
Email: {brechtel | gindele | dillmann}@kit.edu

***Abstract*— Bayesian Occupancy Filtering is an alternative to classical object tracking. Instead of estimating the state of objects in the environment, the latter is separated into equidistant cells. Tracking the occupancy state of these grid-cells is sufficient for many applications in robotics and cell-measurements can be easily produced from almost any kind of sensor. In [6] a sophisticated occupancy filter named BOFUM (Bayesian Occupancy Tracking using prior Map Knowledge) is introduced, which is able to infer velocities solely from occupancy measurements. It also features an advanced process model with motion uncertainty, which can be specialized for different application needs. In this paper we present an approach for recursively applying importance sampling (IS) to approximate the BOFUM calculations. The approach is similar to well known particle filters, but for a discrete cell perspective. In our experiments we achieved a speedup of at least 40-times by using the IS, thus making the algorithm applicable in real-world applications. We evaluate the consequences of approximation in an urban traffic scenario and also show the drawbacks of sampling.**

I. Introduction

Many applications in robotics demand to build a comprehensive model of the real world. Due to incomplete and uncertain perception this turned to one of the key problems in robotics. Especially when dealing with unknown and high numbers of objects, standard approaches, which track objects explicitly, are limited. They have to match object measurements to objects in the state space. When considering all combinations of observations and tracks, the cost of calculation grows exponentially over time with the number of objects. Besides the immense effort that has to be taken to solve the data association problem when many objects are present, every object tracking algorithm needs object detection. This is a requirement that cannot be met in many fields of application and for many types of sensors, especially not when regarding the need for a comprehensive environment model. E.g. in the automotive application presented in this work, we only need a segmentation into free and obstacle space. Grid-based occupancy tracking circumvents suchlike problems, as it simplifies the classical object and track concepts. There is no direct association of measurements and objects. Occupancy measurements only need to be matched to cells, whose positions are certain. Additionally the problem of birth and death of objects is solved implicitly.

The concepts explained here are not constrained to robotics. It is e.g. possible to track toxin flow in the sea for an environmental application with derived filters. In [6] a dynamic occupancy tracker called BOFUM has been introduced, which is able to infer the occupancy's velocity. Thus it is able to predict and track moving traffic participants without ever measuring their velocities or detecting the objects as such. Fig. 2 illustrates the importance of this capability. A screenshot of the tracker applied to laser scanner data is given in fig. 1. Uncertainty in the nonlinear motion model is also modeled (see fig 5b), which makes the filter a well suited base for several robotics tasks like motion planning.

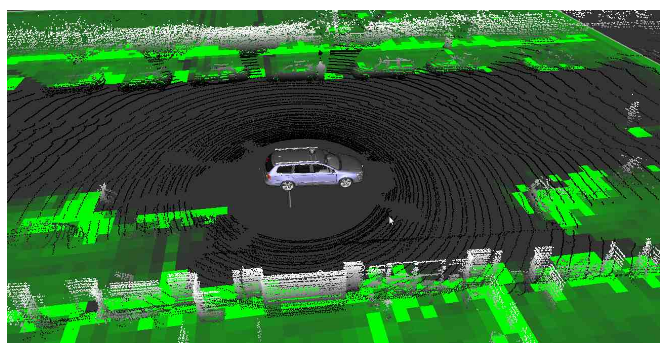


Fig. 1: BOFUM estimation. Measurements are created by the shown Lidar-Scan. Occupancy probability is green.

A shortcoming of this approach is its highly demanding calculation. The complexity of one filter step lies in $O(n^2)$ for n cells, as all cell combinations have to be evaluated. This equals a complexity of $O(r^4)$, when using a square cell field with r rows and columns. Nevertheless the filter's physically motivated process model employing an OR-Relation for the combination of occupancy from different cells (see fig. 5a) favors the application of Monte Carlo methods. In this paper we show how to exploit this by deriving a proposal density for recursive importance sampling (IS). The resulting approach is comparable with Sequential Monte Carlo methods also known as particle filters. One difference in the application of IS is that BOFUM does not essentially need to approximate the posterior, as it could be calculated completely due to its discrete nature. Even so, the filter

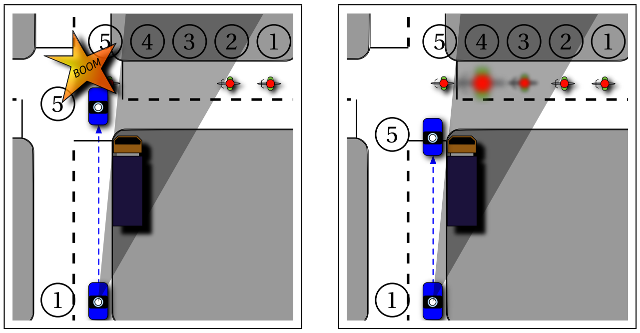


Fig. 2: Example demonstrating the necessity for correct velocity inference in motion planning. The image on the left shows what happens when the object dynamics are ignored. The autonomous car is surprised by the cyclist coming out of the occlusion. In the right image the autonomous car is able to brake, because it *does* predict the cyclists motion, even though it is uncertain.

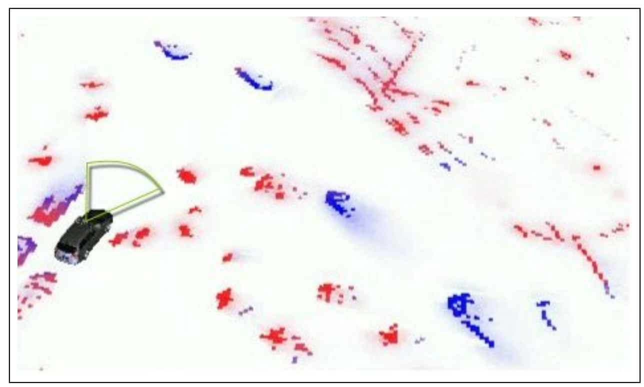
formulation puts itself forward to this kind of calculation speedup. Due to sampling, calculation time no longer directly depends on the grid dimensions, but on the number of samples. The number of necessary samples depends on the variance of the posterior and not on the dimension of the state space [8]. Therefore it is only important how dense the occupation of the space is. This makes three-dimensional occupancy grids possible, because i.e. in aeronautics only small parts of the sky are occupied.

II. Related Work

A. Grid Represenation and Occupancy Filtering

Probabilistic occupancy grids [4], [9] are well-known structures for the description of robot environments and sensor fusion. The space is divided equally in a finite number of rectangular cells on a 2-dimensional plane. Every cell defines a distribution over its possible states, either being occupied or not. To prevent a combinatorial explosion of possible grid configurations in the joint probability distribution, cell states are assumed to be independent from each other. This allows cells to be updated and predicted independently. Their ease of handling and interpretation have made occupancy grids a widely used form of environment representation. They have been successfully applied in the field of mapping and SLAM [13]. Classical occupancy grids are well suited for static environments but are unable to cope with dynamic situations. This nuisance led to the development of dynamic occupancy filters. The work of Prassler et al [10] can be seen as a direct pre-stage of these filters, as they use occupancy grid methods to generate object motion hypotheses. A first attempt to extend occupancy grids with dynamics was done in [5], where the 2-D occupancy grid was extended to a 4-D occupancy grid. Hence measured occupancy had to be associated with a position and speed, which implicitly involves velocity measurements. In [2] a 2-D grid with an additional velocity distribution for each cell was used, making speed sensors optional. This paper builds on BOFUM [6]. Its process model made the velocity not a property of the cell, but of the occupancy inside the cell. The cell grid is here only used as checkerboard, where occupancy can move on. In [1] adaptive motion models were used, by adding movement types in form of object groups as a property of the occupancy in cells. This enables the Bayesian Occupancy Filter using Groups (BOFUG) to infer object classes solely from occupancy measurements. Fig. 3 shows a group estimation result.

(a) Camera image of the scene



(b) Filter Estimation

Fig. 3: BOFUG object group estimation. The camera's field of view is illustrated by the cone. Occupancy classified as *car* is red. The *pedestrian* occupancy is blue. (See [1])

B. Monte Carlo Simulation

Monte Carlo simulations usually aim at drawing samples or approximating properties of a random variable. E.g. the expectation often cannot be calculated analytically, because the so called *target density* $p(x)$ is too complex. It is however possible to evaluate p for any sample x_i. There are many different techniques of how to draw samples, which are more or less sophisticated. A good overview of various sampling methods and their features is given in [8] and [7]. In this work we concentrate on importance sampling (IS), which weights samples drawn from a *proposal density* Q to preserve the expectation. Sequential Monte Carlo methods (SMC) are very popular for classical target tracking and also known as particle filters. Basically they recursively apply IS to approximate an expected object state in a continuous space, even if the process or measurement model are non-linear [3], [12]. Those methods are closely related to the approach presented in this paper, due to methodical similarities. Rao-Blackwellized particle filters close the gap to multiple object tracking by approximating the data-association as well as the birth and death of tracks with SMCs [11].

III. Bayesian Occupancy Filter Using Prior Map Knowledge (BOFUM)

The original occupancy grid algorithms are not sufficient for usage in highly dynamic environments like e.g. traffic scenarios. Propagation models for the occupancy in every grid cell are completely independent in those formulations and moving objects hence can not be described. The BOF approach in [2] extends the state space by the velocity of the occupancy to implement a model of constant speed. The BOFUM algorithm introduced a physically motivated transition model, which further enhances prediction [6]. The huge shortcoming of these dynamic approaches is though, that they square the problem complexity. In opposition to static grids, here combinations of all cells have to be considered for transitions.

A. State Representation

The discrete state space resembles the original BOFUM definitions in [6]:

N: Set containing all n cell indices. The quadratic 2-dimensional cell grid has dimension $\sqrt{n} \times \sqrt{n}$.

$$N = \{1, \dots, n\}$$

O : Vector for occupancy of all cells. If there is at least one object inside the cell boundaries it is declared as *'occupied'*, else as *'not occupied'*.

$$O = \begin{pmatrix} O_1 \\ \vdots \\ O_n \end{pmatrix} \in \{occ, nocc\}^n$$

V: Vector of velocities in x and y- direction. $V_i = (\dot{x}_i, \dot{y}_i)$ is discretized in cells per timestep Δt. Those are the velocities of the occupancy inside the cell, not the velocity of the cell itself. For that reason the velocity distribution of an empty cell carries no information.

$$V = \begin{pmatrix} V_1 \\ \vdots \\ V_n \end{pmatrix} \in \{\mathbb{Z} \times \mathbb{Z}\}^n$$

X: Combination of occupancy and velocity. Hence, it covers all time variant information about the cells at time t.

$$X = (O, V)$$

X^-: Combined state at time $t-1$.

$$X^- = (O^-, V^-)$$

R: Matrix describing whether cell c can be reached from cell a. This information is gained with the help of background knowledge about the current position of the cell grid and map information. Generation of the R matrix is described in [6].

$$R \in \{reach, nreach\}^{n \times n}$$

T: Transition vector. The transition for a cell T_i equals j, if the occupancy in cell i moves to cell j during the next time-step Δt. The transition subsumes velocity, reachability and any other knowledge given about the cell movement. This abstraction allows a flexible integration of context information.

$$T = \begin{pmatrix} T_1 \\ \vdots \\ T_n \end{pmatrix} \in N^n$$

Z: The measurement vector, which includes sensor measurements for every cell. As there is no velocity sensor in the example implementation of this work, the Z_V part of Z will be omitted.

$$\begin{aligned} Z &= (Z_O, Z_V), \\ Z_O &\in \{occ, nocc\}^n, \\ Z_V &\in \{\mathbb{Z} \times \mathbb{Z}\}^n \end{aligned}$$

B. Filter Model and Derivation

In this paper we are not able to explain the BOFUM equations and their origin in full detail. We recommend [6] for a comprehensive description. However, we recall the most important facts in the following.

BOFUM is a classical recursive Bayes Filter and thus separable in a correction and prediction step. It applies the markov assumption for the time-discrete states: A state is supposed to depend only on its preceding state. We derive the filter from the joint composition of all state variables by decomposition:

$$P(X, X^-, R, T, Z) = \prod_{c \in N} \underbrace{P(X_c, X^-, R, T)}_{\text{Prediction}} \underbrace{P(Z_c | X_c)}_{\text{Correction}},$$

where $P(Z_c|X_c)$ specifies the *observation model*. The prediction is decomposed in the following way:

$$P(X_c, X^-, R, T) = P(X^-, R, T) P(X_c | X^-, R, T) = \prod_{i \in N} (O_i^-) \prod_{j \in N} P(T_j, V^-, R) P(O_c | O^-, T) P(V_c | O^-, T)$$

These probability distributions form the BOFUM base. Fig. 4 shows in an overview the conditional dependencies of all variables in form of a Bayesian Network Graph. $\hat{V}_c$ is the expected velocity for cell c and used as an interim variable. For more information see section III-B.2.

While the *observation model* is straight forward and only minor important with respect to IS, especially the prediction and transition models are noteworthy.

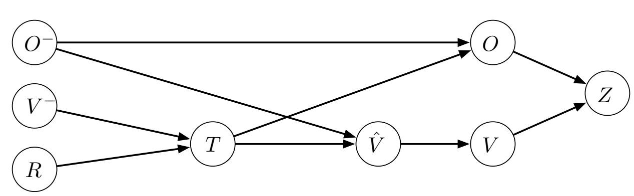


Fig. 4: BOFUM Bayesian Network Graph

1) Occupancy Prediction Model: The combination of cell occupancy is done via disjunction. In former approaches like [2] cell propagations were conditionally independent. Occupancy was therefore combined by averaging. Thereby an occupied cell and an empty cell cancel each other out rather than resulting in an occupied cell. Disjunction guarantees that occupancy moves over empty ground, without being influenced by empty cells. Regarding the sampling this enables us to ignore not-occupied cells during calculation. See fig. 5a for a quick motivation. The lower cells in the sketch do not influence the result of the calculation. This statement can be roughly generalized for uncertain occupancy, as the contribution of a cell to the result of a successor cell directly depends on the cell's occupancy. Apparently it is not completely true, because other occupancy propagated to the target cell reduces a cells influence.

$$\begin{aligned} &P(O_c = occ|O^-, T) \\ &= \begin{cases} 1, & \vee_{a\in N}((O_a = occ) \wedge (T_a = c)) \\ 0, & \text{else} \end{cases} \\ &= 1 - P(O_c = nocc|O^-, T) \end{aligned}$$

A target cell c, given a single antecedent cell a, is occupied, if the transition of a equals c and a itself was occupied.

$$P(O_c = occ|O_a^-, T_a) = \begin{cases} 1, & (O_a = occ) \wedge (T_a = c) \\ 0, & \text{else} \end{cases}$$

2) Velocity Prediction Model: Prediction of velocity works similar. Again, only occupied antecedent cells a with transition index c affect the velocity of the target cell c. As the velocity is assumed to be certain in spatial terms, there is only one antecedent cell for every velocity that can fulfill the requirements. That is why the conditional distribution of the velocity expectation $P(\hat{V})$ can be defined as follows:

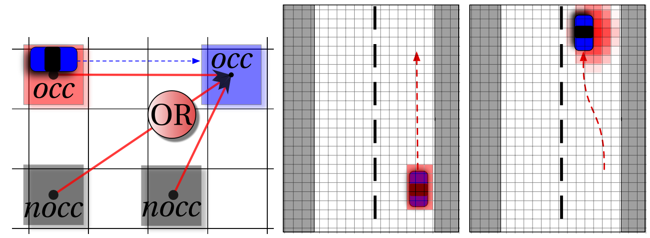
$$P(\hat{V}_c = v(a,c)|O^-, T) = P(O_c = occ|O_a^-, T_a)$$

The helper function $v : N \times N \rightarrow \mathbb{Z} \times \mathbb{Z}$ is used to formulate this concept. It maps a cell index combination to a corresponding velocity.

Uncertainty in acceleration is applied in a separate step after the velocity inference. It subsumes all prediction errors, including the discretization error. This is essential for the filter, otherwise measurements could not be linked to the prediction after unforeseen motions (see fig. 5b). We assume that the error has the characteristics of Gaussian white noise.

$$v_c = \hat{v}_c + w\Delta t \quad \text{with} \quad w \sim \mathcal{N}(0, \Sigma)$$

3) Transition Model: The model for cell transitions, i.e. the model describing the possibility of occupancy movement. This model realizes the principle of *occupancy preservation*, which states that occupancy has to move somewhere and thus can neither spontaneously arise nor disappear. Therefore the coefficient μ_a normalizes over all target cells c of every antecedent a.



(a) Occupancy Combination (b) Acceleration Uncertainty

Fig. 5: Sketches on the combination and prediction of occupancy movements in BOFUM.

$$\begin{aligned} &P(T_a = c, V^-, R) \\ &= \frac{P(T_a = c, V_a^-, R_{a,1}, \dots, R_{a,n})}{\sum_{m\in N} P(T_a = m, V_a^-, R_{a,1}, \dots, R_{a,n})} \\ &= \mu_a P(V_a^-) P(R_{a,c}) P(T_a = c|V_a^-, R_{a,c}) \end{aligned}$$

The conditional dependency formulates that the cells must at the same time be reachable and fit to the occupancy's speed.

$$P(T_a = c|V_a^-, R_{a,c}) = \begin{cases} 1, & V_a^- = v(a,c) \wedge R_{a,c} = reach \\ 0, & \text{else} \end{cases}$$

C. Exact Filter Calculation

Calculation of the posterior is done by marginalization.

$$\begin{aligned} P(X_c|Z) &= \frac{\sum_{X^-,R,T} P(X_c, X^-, R, T, Z)}{\sum_{X^-,R,T,X_c} P(X_c, X^-, R, T, Z)} \\ &\propto \sum_{X^-,R,T} P(X_c, X^-, R, T, Z) \end{aligned}$$

Using the previously presented decomposition, the main stages *Prediction* and *Correction* can be identified. As a third stage, the *Transition* calculation is isolated from the rest of the Prediction.

$$\sum_{O^-,V^-,R,T} P(X_c, X^-, R, T, Z) = \underbrace{P(Z_c|O_c)}_{\text{Correction}} = \underbrace{\sum_{O^-,V^-,R,T} \underbrace{P(V^-, R, T)}_{\text{Transition}} P(O^-) P(O_c|O^-, T) P(V_c|O^-, T)}_{\text{Prediction}}$$

1) Transition Calculation: The transition is calculated in advance, as both the occupancy and velocity prediction depend on it (see fig. 4). This is especially true when applying IS, where the true distribution is approximated through a particle distribution. Exhaustive calculation is done by the following equation:

$$P(T = t) = \tag{1}$$

$$\prod_{a\in N} \mu_a P(V_a^- = v(a, t_a)) P(R_{a,t_a} = reach) \tag{2}$$

2) Velocity Prediction: The interim velocity distribution is inferred with this equation:

$$P(\hat{V}_c = v(a,c)) = \tag{3}$$

$$\sum_{O^-,T} P(O^-)P(T)P(\hat{V}_c = v(a,c)|O^-,T) \tag{4}$$

Finally noise is added to the inferred velocity $\hat{V}$:

$$P(V_c) = \sum_{\hat{V}_c} P(V_c|\hat{V}_c)P(\hat{V}_c) \tag{5}$$

3) Occupancy Prediction: The exact occupation probability of a cell is calculated as follows:

$$P(O_c = occ) \tag{6}$$

$$= 1 - P(O_c = nocc) \tag{7}$$

$$= 1 - \prod_{a \in N} \left(1 - P(O_a^- = occ)P(T_a = c)\right) \tag{8}$$

IV. BOFUM IMPORTANCE SAMPLING

A. Theory

Direct application of importance sampling to the joint cell space is not reasonable, so we concentrate on a single cell view. This is admissible because occupancy transitions are mostly independent from each other. Accordingly P samples are defined as cell transitions:

$$\langle a^{(L)}, c^{(L)} \rangle,\ L \in \{1, \ldots, P\}.$$

The original aim of the occupancy combination is to model, that objects move over empty ground rather than mixing with it. As a side effect cells which are not occupied do not carry any worthy information for the prediction (besides the absence of occupancy), because they cannot influence the prediction results. Omitting the calculation of not-occupied cells does therefore not worsen the filter-result. Similar considerations can be made for the velocity distribution of occupancy inside of cells. This implies that a proper proposal density q should be proportional to the cell's occupancy *and* velocity probability. The combination of antecedent $a^{(L)}$ and target cell $c^{(L)}$ is thus drawn from the following proposal density:

$$\langle a^{(L)}, c^{(L)} \rangle \sim q(a^{(L)}, c^{(L)}) = P(V^-_{a^{(L)}} = v(a^{(L)}, c^{(L)}))P(O^-_{a^{(L)}} = occ)$$

W. l. o. g. we consider in the following only samples $\langle a, c^{L} \rangle$ with the antecedent a. To model the principle of *occupancy preservation*, the BOFUM filter step normalizes over all target cells with the normalizing constant μ_a (refer to (2)):

$$P(T_a = c) = \mu_a\ P(V_a^- = v(a,c))P(R_{a,c} = reach)$$

According to IS, samples have to be weighted with the division $w' = p/q$ of target density p and proposal q to compensate the biased drawing:

$$w'^{(L)} = \frac{P(T_a = c^{(L)})}{q(a, c^{(L)})} = \frac{\mu_a P(R_{a,c^{(L)}} = reach)}{P(O_a^- = occ)}$$

Thus the time-consuming calculation of $P(V_a^- = v(a,c))$, basically an approximation of an integration over several bivariate Gaussian distributions, can be omitted as we can directly draw from the Gaussian distribution (see equation (5)). The obligatory normalization over all samples $\langle a, c^{(M)} \rangle$ eliminates the normalization coefficient μ_a of the occupancy preservation:

$$w^{(L)} = \frac{P(T_a = c^{(L)})}{\sum_M P(T_a = c^{(M)})}$$

Since we only consider samples with antecedent a the a priori occupancy probability $P(O_a^- = occ)$ is for all the same and therefore left out. It only influences the sampling density of the antecedents not their weights. Particle weighting for the transition approximation results in the following equation:

$$w^{(L)} = \frac{P(R_{a,c^{(L)}} = reach)}{\sum_{\langle a, c^{(M)} \rangle} P(R_{a,c^{(M)}} = reach)} = \mu'_a P(R_{a,c^{(L)}} = reach)$$

Hence as weighting factor only the reachability remains. The proposal density can be improved by additionally sampling from the reachability. In our implementation this is not reasonable, since the calculation of the complete R matrix is computationally demanding.

We deduce the noise-free velocity $\hat{V}$ from the approximated transition according to (4):

$$P(\hat{V}_{c^{(L)}} = v(a, c^{(L)})) = P(O_a^- = occ)\ w^{(L)}$$

$P(\hat{V}_c)$ equals zero, if no appropriate transition sample was drawn. For the occupancy prediction (8) we can proceed analogously and reuse the previously evaluated velocity prediction:

$$P(O_c = occ) = 1 - \prod_{a \in N} \left(1 - P\left(\hat{V}_{c^{(L)}} = v(a, c^{(L)})\right)\right)$$

At first sight it might be impractical to choose a proposal density proportional to the cell-occupancy when approximating the transition, because the a priori occupancy has no impact on the transition result at all. However, with respect to approximating the a posteriori velocity and occupancy this choice is straight forward. It raises the sample density in regions, that have high impact on the expectation calculation of the posterior. Thus the approximation quality is improved.

While the proposal density leads to an optimal approximation for velocity inference – when noticing that reachability and measurement can due to their nature not be incorporated in the sampling process – this is not true for occupancy inference. The conditional dependency between transitions in form of the OR-relation prevents that. It is however the best starting density, when not sampling from the complex joint distribution and has the inevitable advantage of reusing the velocity result as is.

V. EXPERIMENTS

In this section we will first show the effects of the IS approximation on the filter result in a worst case scenario. This gives a clue about the minimal number of samples needed to obtain appropriate approximation results. Second an analysis for a recorded traffic scene is done, to show the hazards and chances of approximation in real world applications. Finally a short runtime overview is given to demonstrate the benefits of the sampling algorithm.

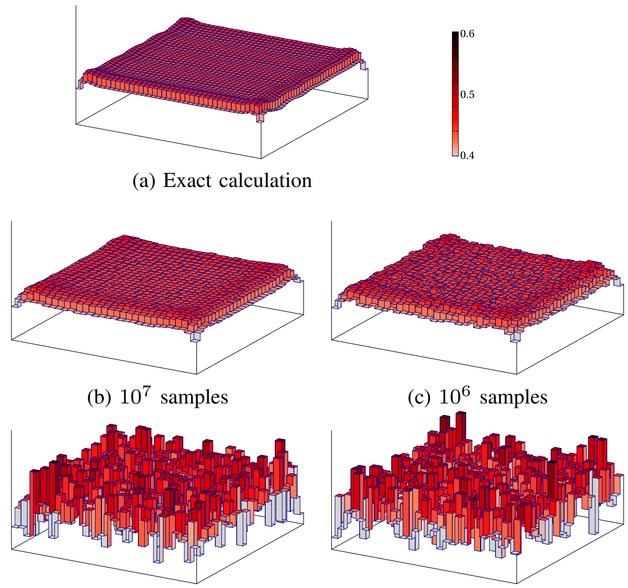
A. *Worst Case Analysis*

This experiment is meant to reveal the lower boundary for the sampling density. Note that it is actually not the worst case, but a very unfavorable prior, that will never be deceeded in reality. The grid size is 30×30 and contains cells of 1 m width. The initial occupancy distribution in this experiment is uniform and thus unfavorable regarding the sampling strategy. Interaction between different transitions occurs very often, which lowers the quality of the proposal density, because it assumes independent transitions (cf. IV-A). Additionally the a priori velocity distributions have a standard deviation of $4\,\frac{\text{m}}{\text{s}^2}$. The filtering frequency is 1 Hz, thus the high variance spreads the occupancy very far over the grid in one timestep. High variance of the proposal density naturally raises the need for more samples.

The results of the experiments in fig. 6 demonstrate the weaknesses of sampling algorithms. The correct calculation only needs to pursue 810,000 cell transitions. Even when using the ridiculous high number of 10 million samples, one can see approximation errors. Here the exact calculation is superior to the approximation in terms of speed, because it calculates every cell combination only once. Practically the result for 1 million samples is already usable. This shows that the sampling works even in bad conditions nearly as well as the complete calculation. The flattened borders form, because there is no occupancy movement originating outside the grid.

B. *Real Case Analysis*

In the worst case analysis IS is indeed slower for a reasonable number of particles. In real scenarios however, it goes strong. The experiment in fig. 7 demonstrates its qualities: Here a 80×80 is used to filter measurements from a Lidar Velodyne HDL-64E 360° laser scanner. The cell width is chosen to be practically usable 0.5 meters. To obtain the exact result of a filter step, one would have to calculate 41 million cell transitions. The sampling algorithm creates very good results with only 1 million sample transitions. Because the runtime is approximately proportional to the number of samples, we conclude that in this scenario IS is at least 40-times faster than the exact calculation. When defining usability by still recognizing all objects, even 10^5 samples are sufficient. This can be exemplarily seen for the very small cyclist on the left side of the scene. Very low numbers of samples however suffer from noise in occluded and therefore highly uncertain areas. Sampling allows to arbitrarily shift between exactness and sampling time according to one's requirements. For an application aiming at 10 Hz a grid resolution of 160×160 is realistic, when using the current implementation.



(a) Exact calculation

(b) 10^7 samples (c) 10^6 samples

(d) 10^5 samples (e) 10^4 samples

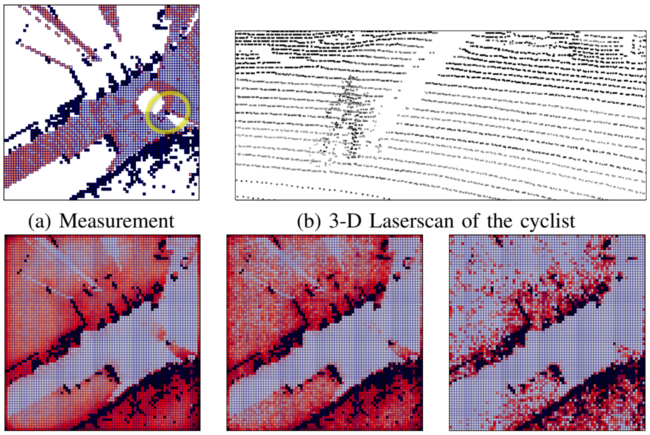
Fig. 6: Prediction results for the approximation in a worst case scenario. The calculation time for this experiment is shown in figure 8. Note that the scale is magnified to make the errors better visible.

C. *Runtime and Speedup*

The runtime analysis for both the worst and real case discussed above is shown in fig. 8 in percents of the exact worst case runtime. The latter needs about 50 milliseconds using an Intel® Core™ i7 920 based computer. The results were measured by counting the CPU cycles to eliminate external factors. Calculation effort for the approximation scales almost linear with the number of samples even for extreme sample numbers. The complexity of the algorithm is no longer quadratic to the number of cells, but depends on the amount of occupancy in a scene and the variance of its velocities. Advantages due to the simpler nature of the approximated calculation cannot be seen in these results. An efficient method for approximating Gaussian integrals equalizes this disadvantage of the exact calculation.

VI. CONCLUSION

In this paper we presented an efficient method for approximative grid based occupancy tracking. We utilize importance sampling to tremendously cut down calculation time for prediction and thus make BOFUM real-time capable. The speedup due to the approximation totals more than 4000% compared to the exact calculation time in our experiment. It also allows to adaptively trade off speed and accuracy according to the application's needs, thereby providing anytime characteristics. Additionally the independent sampling procedure allows to easily parallelize the filtering.



(a) Measurement (b) 3-D Laserscan of the cyclist

(c) 10^6 samples (d) 10^5 samples (e) 10^4 samples

Fig. 7: A posteriori estimation of BOFUM for different sample counts for real measurements. The cyclist is highlighted by the yellow ellipse.

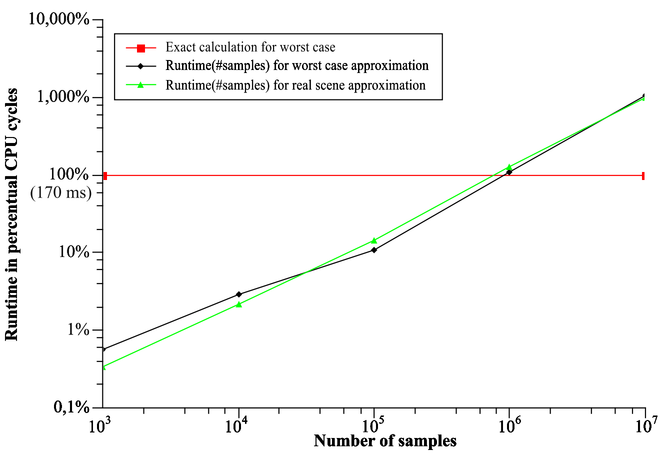


Fig. 8: Average cpu cycles for one predict step calculation in the worst case and real case scenario shown in fig. 6 and 7 for different numbers of samples. The results are given in percent of the worst case cycle number.

Further we are working on a combination of the *tracks* concept used in target tracking and the cell-based occupancy concept. In our opinion fusing the rather sensor independent BOFUM with object specific estimations, that are able to integrate very specific object knowledge, is the way to go. Therefore we try to integrate BOFUM in a sophisticated multi hypothesis tracking (MHT). Recent research has shown that the inference of different object classes is possible on occupancy level, solely by using multiple transition models [1]. Only modest changes to BOFUM were needed to make it automatically segment the environment into different occupancy groups. Thanks to the sampling technique presented in this paper performance does not decrease, although the state space grows proportional to the number of groups. This is due to the orthogonal nature, the different group motion models were chosen from. The target tracking algorithm will be integrated in a similar manner.

Thus we think that Bayesian Occupancy Filtering provides an outstanding platform for comprehensive environment estimation and prediction using several different sensors as information source.

ACKNOWLEDGMENT

The authors gratefully acknowledge the contribution of the German collaborative research center "SFB/TR 28 – Cognitive Cars" granted by Deutsche Forschungsgemeinschaft.

REFERENCES

[1] Sebastian Brechtel, Tobias Gindele, Jan Vogelgesang, and Rüdiger Dillmann. Probabilistisches Belegtheitsfilter zur Schätzung dynamischer Umgebungen unter Verwendung multipler Bewegungsmodelle. In *Proceedings of the 21th Fachgespräch Autonome Mobile Systeme*, pages 49–56, 2009.

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

[3] Arnaud Doucet, Simon Godsill, and Christophe Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.

[4] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57, 1989.

[5] C. Fulgenzi, A. Spalanzani, and C. Laugier. Dynamic Obstacle Avoidance in uncertain environment combining PVOs and Occupancy Grid. *Robotics and Automation, 2007 IEEE International Conference on*, pages 1610–1616, 2007.

[6] T. Gindele, S. Brechtel, J. Schröder, and R. Dillmann. Bayesian Occupancy Grid Filter for Dynamic Environments Using Prior Map Knowledge. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, Xi'an, China, Juni 2009.

[7] M.H. Kalos and P.A. Whitlock. *Monte carlo methods*. WILEY-VCH Verlag GmbH,Weinheini,Germany, 2008.

[8] David J. C. Mackay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press,United Kingdom, March 2005.

[9] H.P. Moravec. Sensor Fusion in Certainty Grids for Mobile Robots. *AI Magazine*, 9(2):61, 1988.

[10] E. Prassler, J. Scholz, and A. Elfes. Tracking Multiple Moving Objects for Real-Time Robot Navigation. *Autonomous Robots*, 8(2):105–116, 2000.

[11] Simo Särkkä, Aki Vehtari, and Jouko Lampinen. Rao-Blackwellized particle filter for multiple target tracking. *Information Fusion*, 8(1):2–15, 2007.

[12] S. Thrun. Particle filters in robotics. *Proceedings of the 17th Annual Conference on Uncertainty in AI (UAI)*, 1, 2002.

[13] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. MIT press, Cambridge, Massachusetts, USA, 2005.