Detecting Changes in the Environment Based on Full Posterior Distributions over Real-valued Grid Maps

Lukas Luft, Alexander Schaefer, Tobias Schubert, Wolfram Burgard

***Abstract*—To detect changes in an environment, one has to decide whether a set of observations is incompatible with a set of previously made observations. For binary, lidar-based grid maps, this is essentially the case when the laser beam traverses a voxel which has been observed as occupied, or when the beam is reflected by a voxel which has been observed as empty. However, due to discretization errors, some voxels are neither completely occupied nor completely free. These voxels have to be modeled by real-valued variables, whose estimation is an inherently statistical process. Thus, it is nontrivial to decide whether two sets of observations emerge from the same underlying true map values, and hence from an unchanged environment. Our main idea is to account for the statistical nature of the estimation process by leveraging the full map posteriors instead of only the most likely maps. Closed-form solutions of posteriors over real-valued grid maps have been introduced recently. We leverage a similarity measure on these posteriors to calculate for each point in time the probability that it constitutes a change in the hidden map value. While the proposed approach works for any type of real-valued grid map that allows the computation of the full posterior, we provide all formulas for the well-known reflection maps and the recently introduced decay-rate maps. We introduce and compare different similarity measures and show that our method significantly outperforms baseline approaches in simulated and real world experiments.**

***Index Terms*—Mapping, range sensing**

Fig. 1: Exemplary views of the changing environment in which we conduct one of our mapping experiments.

I. Introduction

Grid maps are a popular approach in the context of mobile robots to represent the environment. Their key idea is to partition the environment into discrete voxels, where each voxel stores a particular value related to a specific property of the environment in the corresponding location. Lidars are a popular sensor for building such maps due to their accuracy. They send out laser beams and measure the time the beams need to return after being reflected from objects in the environment.

Typically, environments exhibit temporary or permanent changes, which the robot has to detect and react to. Consider occupancy grid maps, in which each cell is either free or occupied. Here, observing a reflection in a free cell or a transmission through an occupied cell is an indication of a change in the environment. However, reasoning about these changes is more complicated in real-world applications, where cells are not always completely free or occupied. These types of cells are required to represent semi-transparent objects, structures smaller than the voxel size, or glass. To model the stochastic behavior of a beam within such a voxel, one has to use real-valued maps. Their values can either be interpreted as occupancy probabilities [1], [2], [3], reflection probabilities [4], or as expected laser ray lengths [5]. It is non-trivial to decide whether a set of observations is within stochastic fluctuation or due to an actual change in the environment. This is particularly true if the map representation only comprises the most likely values, as opposed to the full posterior distributions. Consider, for example, a set of observations within a voxel, most of which report a reflection. Only with a notion of uncertainty about a previously estimated map value, one can judge whether the observations are likely to be in accordance with this value.

Our main idea is that if the full map posteriors are given, we can score pairs of temporary maps conditioned on distinct sets of observations according to their compatibility. To get the map posteriors we apply mapping with known poses. Luft et al. [6] recently introduced closed-form solutions to the full map posteriors over real-valued maps. For each point in time, this approach computes two posteriors, the one conditioned on all measurements after this point in time and the one

This work has been partially supported by the European Commission in the Horizon 2020 framework program under grant agreements 644227-Flourish and 645403-RobDREAM, by the Graduate School of Robotics in Freiburg, and by the State Graduate Funding Program of Baden-Württemberg. All authors are with the Department of Computer Science, University of Freiburg, Germany. {luft,aschaef,tobschub,burgard}@cs.uni-freiburg.de

conditioned on all measurements before it. We leverage a similarity measure on these posteriors to score each point in time according to the probability that it constitutes a change in the hidden map value for each voxel.

The proposed framework applies to any measurement model that allows to calculate the full posterior over a real-valued map, for example the well-known reflection model [4] and the recently introduced decay-rate model [5]. The input to the proposed algorithm is the ordered sequence of observations. For the reflection model, the elements of this stream contain the counts of how often beams were reflected by a cell or crossed it. For the decay-rate model, they consist of the numbers of reflections and the total distances that all beams travelled within each cell during the mapping process.

Conceptually, there are three main features that set the proposed method apart from other approaches: It considers the full path information, as opposed to end-point models; It considers real valued maps, as opposed to occupancy maps; It uses the full posterior over these real map values. Please note that previous approaches, for example the one proposed by Arbuckle et al. [7], use real valued posteriors over binary occupancy maps, which is different from using full posteriors over real map values.

II. Related Work

There has been a substantial amount of research on grid mapping in dynamic environments. Fox et al. [8] propose two methods to discard measurements in the localization process that are unlikely to be due to mapped objects. One of these methods is the entropy filter. It accounts for only those measurements that confirm the robot's belief – which can lead to over-confidence. As opposed to that, our method always accounts for the whole stream of measurements after a predicted change in the environment. The second method which Fox et al. describe [8] – the distance filter – discards measurements with ranges smaller than predicted from the current map. While the latter is suited for the case that unexpected objects appear, it can not handle changes where objects are removed from the scene. In contrast, our approach is not only able to detect voxels that turn from occupied to free and vice versa, but also reasons about changes of non-binary map values.

Hähnel et al. [4] introduce a hidden variable that indicates whether a measurement is caused by a dynamic object. In contrast to our method, both Fox et al. [8] and Hähnel et al. [4] treat dynamic objects as outliers instead of adjusting the map accordingly.

Other approaches incorporate information about the dynamics within the map. Arbuckle et al. [7] store multiple occupancy values estimated over different timescales in so-called Temporal Occupancy Grids. While this introduces additional parameters and memory consumption, our approach keeps track of only the most recent map.

Luber et al. [9] model the occurrence of humans by a Poisson process. They store the learned parameter – the temporal rate at which a person appears at a certain point in space – in a so-called affordance map. Saarinen et al. [10] extend this approach by learning two rates: the one at which occupied cells turn free, and vice versa. Additionally, they use a weighting method to prefer recent measurements over older ones. Meyer-Delius et al. [11] extend the Bayesian update formula for the occupancy posterior by a state transition probability. The latter term is not given a priori but estimated during operation. Compared to the these previous methods, our approach is more general in the sense that it forgoes an explicit model of the underlying dynamics.

Biber and Duckett [12] maintain multiple maps learned from data on different timescales. Similar to Yamauchi and R. Beer [13], they use recency-weighted averaging to suppress old measurements. In their approach, the decay-rate of the weighting term depends on the timescale. In both approaches, the weighting terms have to be fixed a priori while our algorithm estimates the breakpoint for each region individually.[1]

All approaches discussed so far either assume a binary occupancy state or work with the most likely maps only instead of the full posterior distribution. As opposed to that, we leverage the full posterior over real-valued maps, which we consider to be the most significant difference to all approaches discussed in this section.

Fehr et al. [14] conduct RGB-D-based 3D reconstructions in dynamic environments with a segmentation of dynamic objects. Krajnik et al. [15] model the environment, e.g., the occupancy states of a grid, as periodically changing in time and store the most prominent Fourier coefficients. Andreasson et al. [16] calculate the probability for each reflection point in a laser scan to be different from a previously recorded reference model. The latter is represented as a 3D grid comprising the mean and covariance values obtained from the Normal Distribution Transformation (NDT) of the initial laser point cloud. As opposed to our approach, Andreasson et al. [16] do not account for the whole path information of the laser beams but only for their endpoints.

Besides the grid-based approaches, there are also feature-based methods that consider dynamic objects, for example the work of Andrade-Cetto and Sanfeliu [17], Sofman et al. [18], and Manso et al. [19].

III. Approach

In this section, we first define how we formalize a change in the map. We then briefly describe the basics of posterior distributions over real valued grid maps [6] and provide the formulas needed for the remainder of the paper. After we formulate the search for a point in time at which the map changes as minimization of certain measures on map posteriors, we introduce three different options for these measures in sections III-A to III-C.

Similar to most other grid-based mapping approaches, we model the map value of each individual voxel as hidden random variable $\mathcal{M}$ and estimate its value from a set of n observations Z. The posterior distribution $\mathrm{bel}^Z(m) := p(m \mid Z)$ of the map value $m \in \mathcal{M}$ given all measurements Z is called belief.

[1] Our algorithm uses a measure of compatibility between map posteriors. This measure is interchangeable. One of the three proposed measures also contains an a priori fixed threshold, see III-C.

Let us assume that the environment changes at some point in time (let us call it a breakpoint b) because an object is removed or added. Then, instantly, the hidden variables corresponding to voxels that are affected by this change are replaced by new variables. For the voxels in question, all measurements taken before the breakpoint bias the estimate of the new random variable. Thus, for a potential breakpoint, we define two distributions: the map posterior $\mathrm{bel}^{1:b-1}$ conditioned on the measurements before the breakpoint and the posterior $\mathrm{bel}^{b:n}$ conditioned on the recent measurements from the breakpoint on. If the robot has to localize itself, it must use the most recent map. Thus, it is desirable to detect the breakpoint b and maintain $\mathrm{bel}^{b:n}$.

Luft et al. [6] show that, for two particular measurement models, the beliefs over individual non-binary map values can be parametrized as follows. The well-known reflection model [4] assigns to each voxel a reflection probability μ. Given the hidden map value μ, the likelihood to observe a particular stream Z of observations with H reflections (hits) and M transmissions (misses) in the cell is

$$L_\mu(Z) := p(Z \mid \mu) = \mu^H \left(1-\mu\right)^M. \tag{1}$$

The map posteriors are beta-distributions [6]

$$\mathrm{bel}(\mu) = \mathrm{Beta}(\mu;\alpha,\beta) = \frac{\mu^{\alpha-1}(1-\mu)^{\beta-1}}{\mathrm{B}(\alpha,\beta)}, \tag{2}$$

with the beta function $\mathrm{B}(\cdot,\cdot)$. The parameters α and β are determined by the number of reflections and transmissions: $\alpha = H + \alpha_0$ and $\beta = M + \beta_0$, where α_0 and β_0 are priors and $\alpha_0 = \beta_0 = 1$ for a uniform prior. The recently introduced decay-rate model [5] assigns each voxel a decay-rate λ. Here, the likelihood for a stream Z with H reflections and all laser beams travel a total distance R within the cell is

$$L_\lambda(Z) := p(Z \mid \lambda) = \lambda^H \exp\left(-\lambda R\right). \tag{3}$$

The map posteriors are gamma-distributions [6]

$$\mathrm{bel}(\lambda) = \mathrm{Gamma}(\lambda;\alpha,\beta) = \frac{\beta^\alpha}{\Gamma(\alpha)}\lambda^{\alpha-1}e^{-\beta\lambda}, \tag{4}$$

with the gamma function $\Gamma(\cdot)$, $\alpha = H + \alpha_0$, $\beta = R + \beta_0$, where $\alpha_0 = 1$ and $\beta_0 = 0$ for an uninformed prior.

Please note that in their original formulation [1], [2], [3], occupancy grid maps possess binary map values, such that the map posterior for each voxel is a Bernoulli distribution, represented by one real value p. As opposed to that, for the two map representations introduced in this section, the map values are already real numbers, such that the posterior for each voxel is a continuous distribution.

Based on the parametrized posteriors, the aim of this paper is to formalize a criterion that allows a breakpoint detection. We are looking for a measure M such that the expected breakpoint b^* is

$$b^* = \underset{b \in B}{\operatorname{argmin}}\ \mathrm{M}\left(\mathrm{bel}^{1:b-1}, \mathrm{bel}^{b:n}\right), \tag{5}$$

where $B \subset \mathbb{N}^{\leq n}$ is a set of potential breakpoints. Please note that $b^* = 1$ means that we detect that the environment did not change. We ensure that $1 \in B$, where $\mathrm{bel}^{1:0} = \mathrm{bel}^{\emptyset}$ is a prior distribution. To associate each of the collected measurements to either $\mathrm{bel}^{1:b-1}$ or $\mathrm{bel}^{b:n}$ we have to preserve their temporal order. However, the order between potential breakpoints can be neglected to the benefit of reduced memory usage, which is particularly relevant for $|B| \ll n$. This applies for lidars that see a voxel multiple times within one scan almost simultaneously, assuming that a breakpoint can only occur between individual scans. Therefore, in our real-world experiments IV-B, we define the potential breakpoints to lie between individual scans. Depending on the concrete use case, one can choose the potential breakpoints substantially more sparsely. A service robot, for example, could check for breakpoints every time it revisits a particular room. Accordingly, it is sufficient to store the map generated during a particular visit as a single measurement, instead of maintaining the entire stream of laser scans. Please note that our approach explicitly assumes instantaneous changes of map values. This is no loss of generality, as one can choose the set of possible breakpoints such that it fits the dynamics of the situation.

In principle, it is possible to apply our framework in an online setting. Every time a new laser scan arrives, the map posteriors are updated recursively and the values M in (5) have to be recalculated. When a breakpoint is detected, all previous measurements can be deleted.

Thus far, we assumed that each voxel contains either a single breakpoint or no breakpoints at all. However, in many applications the robot might face multiple changes within individual voxels during its mission. Ideally, these breakpoints manifest as local minima in the objective function M in (5). The experiments in Section IV-A indicate that our method can also deal with multiple breakpoints. However, it is subject of future research to take these multiple breakpoints more explicitly into account.

In the remainder of this section, we discuss three different options for the measure M in (5).

A. Bayesian Information Criterion

A natural choice for M is the Bayesian Information Criterion (BIC) [20]

$$\mathrm{BIC} = \ln(n)\,k - 2\ln(L), \tag{6}$$

with the number of measurements n, the number of parameters in the model k, and the maximized measurement likelihood L. The first term penalizes over-fitting, while the second term rewards the goodness of fit.

For $b = 1$, we assume that we only have one map value. Then, the likelihood function for the reflection model is given by (1) and the likelihood for the decay-rate model is given by (3). These likelihoods both possess only a single parameter, namely μ or λ, respectively. Thus, we set $k = 1$ in (6). If we assume two posteriors – one conditioned on the measurements before b and the other conditioned on the measurements after – we have $k = 3$. The parameters are the breakpoint b and the map values of both posteriors. For the reflection model, the likelihood term can thus be calculated from (1) as

$$L = L_{\mu^{1:b-1}}\left(Z^{1:b-1}\right) L_{\mu^{b:n}}\left(Z^{b:n}\right),$$

with the maximum likelihood parameters

$$\mu^{1:b-1} = H^{1:b-1}(H^{1:b-1} + M^{1:b-1})^{-1}$$

and $\mu^{b:n}$ accordingly [6]. For the decay-rate model, the likelihood term can be calculated from (3) as

$$L = L_{\lambda^{1:b-1}}\left(Z^{1:b-1}\right) L_{\lambda^{b:n}}\left(Z^{b:n}\right),$$

with the maximum likelihood parameters

$$\lambda^{1:b-1} = H^{1:b-1}(R^{1:b-1})^{-1}$$

and $\lambda^{b:n}$ accordingly [6].

Please note that the map posteriors are sufficient to calculate (6). Thus, minimizing the BIC is a special case of (5). If the minimum is attained at $b = 1$, the BIC-based method implies that there is no change in the environment.

In the following, we introduce two measures that consider the full map posteriors in contrast to the BIC, which only accounts for the measurement likelihood given the most likely map parameters,

B. Entropy-based approach

As a second option for M we introduce an entropy-based approach. We assume that the entropy of a posterior distribution with given parameterization generally decreases with the number of measurements. This is implies that if the incorporation of an additional set of measurements leads to an increase of the entropy, this set must be generated from a different random variable. Thus we are looking for the set of most recent measurements that minimize the entropy

$$b^* = \underset{b \in B}{\operatorname{argmin}}\ \mathrm{H}\left(\mathrm{bel}^{b:n}\right). \tag{7}$$

For the reflection model, the posterior $\mathrm{bel}^{b:n}$ is a Beta distribution. Its entropy is

$$\begin{aligned}\mathrm{H}\left[\mathrm{Beta}(\alpha,\beta)\right] =& \ln\left[\mathrm{B}(\alpha,\beta)\right] + (\alpha+\beta-2)\psi(\alpha+\beta)\\ &- (\alpha-1)\psi(\alpha) - (\beta-1)\psi(\beta)\end{aligned}$$

with the digamma function $\psi(\cdot)$ and the beta function $\mathrm{B}(\cdot,\cdot)$. For the decay-rate model, the posterior $\mathrm{bel}^{b:n}$ is a Gamma distribution with entropy

$$\mathrm{H}\left[\mathrm{Gamma}(\alpha,\beta)\right] = \alpha + \ln\left(\frac{\Gamma(\alpha)}{\beta}\right) + (1-\alpha)\psi(\alpha)$$

where $\Gamma(\cdot)$ is the gamma function.

Please note that in the case of binary occupancy grids, the entropy of the posterior is minimal for the Bernoulli parameters $p = 0$ or $p = 1$. Thus, the entropy-based approach always maintains the latest stream of observations which contains either only hits or only misses. As this will lead to severe over-confidence, it is essential to leverage the posteriors over real-valued maps.

C. Probabilistic approach

In addition to the BIC, which considers the most likely map parameters, and the entropy-based approach, which accounts for the full map posterior conditioned on the measurements after a potential breakpoint, we now introduce a new measure, which compares two map posteriors. Therefor, we analytically derive the probability density P_b that two posterior distributions $\mathrm{bel}^{1:b-1}$ and $\mathrm{bel}^{b:n}$ are generated from the same underlying random variable:

$$\begin{aligned}P_b :=& \int_m \int_{m'} \mathrm{bel}^{1:b-1}(m)\,\mathrm{bel}^{b:n}(m')\delta(m-m')dmdm'\\ =& \int_m \mathrm{bel}^{1:b-1}(m)\,\mathrm{bel}^{b:n}(m)dm.\end{aligned} \tag{8}$$

We assume that the probability for both distributions to be generated by the same hidden variable is minimal at the breakpoint. Thus

$$b^* = \underset{b \in B}{\operatorname{argmin}}\ P_b\left(\mathrm{bel}^{1:b-1}, \mathrm{bel}^{b:n}\right). \tag{9}$$

For the reflection model, the objective function reduces to

$$\begin{aligned}P_b =& \int_0^1 \mathrm{Beta}\left(\mu;\alpha^{1:b-1},\beta^{1:b-1}\right)\mathrm{Beta}\left(\mu;\alpha^{b:n},\beta^{b:n}\right)d\mu\\ =& \int_0^1 \frac{\mu^{\alpha^{1:b-1}-1}(1-\mu)^{\beta^{1:b-1}-1}}{\mathrm{B}(\alpha^{1:b-1},\beta^{1:b-1})}\frac{\mu^{\alpha^{b:n}-1}(1-\mu)^{\beta^{b:n}-1}}{\mathrm{B}(\alpha^{b:n},\beta^{b:n})}d\mu\\ =& \int_0^1 \frac{\mu^{\alpha^{1:b-1}+\alpha^{b:n}-2}(1-\mu)^{\beta^{1:b-1}+\beta^{b:n}-2}}{\mathrm{B}(\alpha^{1:b-1},\beta^{1:b-1})\,\mathrm{B}(\alpha^{b:n},\beta^{b:n})}d\mu\\ =& \int_0^1 \mathrm{Beta}(\mu;\alpha^{1:b-1}+\alpha^{b:n}-1,\beta^{1:b-1}+\beta^{b:n}-1)d\mu\\ &\cdot\frac{\mathrm{B}(\alpha^{1:b-1}+\alpha^{b:n}-1,\beta^{1:b-1}+\beta^{b:n}-1)}{\mathrm{B}(\alpha^{1:b-1},\beta^{1:b-1})\,\mathrm{B}(\alpha^{b:n},\beta^{b:n})}\\ =& \frac{\mathrm{B}(\alpha^{1:b-1}+\alpha^{b:n}-1,\beta^{1:b-1}+\beta^{b:n}-1)}{\mathrm{B}(\alpha^{1:b-1},\beta^{1:b-1})\,\mathrm{B}(\alpha^{b:n},\beta^{b:n})}\end{aligned} \tag{10}$$

For the decay-rate model, we replace the beta distributions by gamma distributions and integrate from zero to infinity. With a similar derivation as in (10), we get

$$\begin{aligned}P_b =& \left(\frac{\beta^{1:b-1}}{\beta^{1:b-1}+\beta^{b:n}}\right)^{\alpha^{1:b-1}}\left(\frac{\beta^{b:n}}{\beta^{1:b-1}+\beta^{b:n}}\right)^{\alpha^{b:n}}\\ &\cdot\frac{(\beta^{1:b-1}+\beta^{b:n})}{(\alpha^{1:b-1}+\alpha^{b:n}-1)\,\mathrm{B}(\alpha^{1:b-1},\alpha^{b:n})}\end{aligned} \tag{11}$$

Please note that P_b is a particular value of a probability density function. To determine the value P_1 from (8), one has to define $\mathrm{bel}^{\emptyset}$. In our experiments, we treat P_1 as a fixed parameter and determine it via cross-validation, as explained in Section IV-A.

It is straightforward to apply the metric (8) to binary occupancy maps where $m \in \{0,1\}$. Here, the real value μ is the occupancy probability and hence represents the full posterior. Replacing the integral in (8) by a sum, yields

$$P_b = \mu^{1:b-1}\mu^{b:n} + \left(1-\mu^{1:b-1}\right)\left(1-\mu^{b:n}\right). \tag{12}$$

Here, P_b is a probability, as opposed to the probability density function in the case of the real-valued maps. A natural choice of the prior $\mu^{\emptyset} = 0.5$ results in $P_1 = 0.5$.

IV. EXPERIMENTS

To evaluate our method, we perform experiments in simulation and with a real-world dataset. In the experiments, we compare the following approaches

- TRUE: uses the true breakpoint.
- BIC: the Bayesian Information Criterion III-A.
- ENT: the entropy-based approach III-B.
- PRO: the probabilistic approach III-C.
- BIN: the probabilistic approach on binary maps (12).
- NDT: the NDT-based approach [16].
- BASE: the baseline that assumes a static environment ($b = 1$).

Note that TRUE is the map generated from all measurements after the true (typically unknown) breakpoint. We compare our method against NDT as presented by Andreason et al. [16]. It generates a reference model of the initial environment represented by the Normal Distribution Transformation of the corresponding point cloud. For each point in the subsequent laser scans, it computes the probability to be different from this reference model.

A. Simulation experiments

We perform an experiment with randomly chosen ground truth map values and simulated lidar observations as follows. We consider an environment of $N = 10^4$ voxels. A robot visits every voxel n times and collects the measurements H, M, and R. We simulate these measurements according to randomly generated ground truth map values μ and λ. For each cell, we draw μ and $P_{ref} := 1 - \exp(-\lambda \cdot l)$ with $l = 1m$ from a uniform distribution over $[0, 1]$.[2] To simulate changes in the map, these true map values change after a breakpoint b. For each voxel, we draw b from a uniform distribution over $\{1, \dots, n\}$. For each voxel, we estimate the breakpoint based on the different approaches and use the stream of all measurements from b on to generate the map posterior. We repeat the whole experiment for $n \in \{5, 10, 20, 50, 100, 200, 500\}$. Figs. 2c and 2d show the root mean squared error per voxel (RMSE) between the maps estimated by the different approaches and the true map. For the decay-rate maps, we compare P_{ref} instead of λ.

The algorithm produces a false positive if it detects a breakpoint where the true map value remains the same. These false positives are particularly adverse in static environments, where a false breakpoint detection reduces the number of measurements taken into account for mapping. The situation most prone to false positives is when all map values are constant. The results are shown in Figs. 2e and 2f. Note that in this case the baseline approach coincides with the approach based on the true breakpoint.

To illustrate our intuition that the proposed algorithm can also handle multiple changes within one voxel, we run the simulation again with multiple breakpoints per voxel. For each voxel, we draw the number of breakpoints from a uniform distribution over the integers one to five. The corresponding results are shown in Figs. 2a and 2b.

Please note that the value P_1, which we need for the probabilistic approach, cannot be directly computed from (8). In the experiments, we determine it via cross-validation. Therefor, we run the whole simulation experiment including the changing and the stationary scenario for different values of P_1. We choose the value of P_1 that minimizes the total RMSE on this training phase and conduct the evaluation in Fig. 2 on an independent simulation.

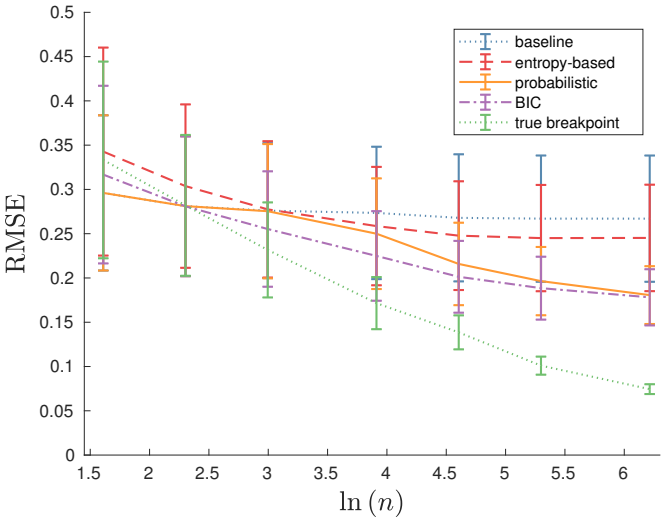
For both measurement models and all approaches except for the baseline, the error decreases with the number of collected measurements. In the changing environment with the decay-rate model, we observe that BIC and PRO perform better than ENT. For the reflection model, this effect manifests only with a growing number of measurements ($n > 20$). We attribute this to the fact that the decay-rate model takes into account the real valued distances of beams within a voxel in addition to the integer hits and misses. Therefore, it contains more information than the reflection model, which is particularly relevant for small n. Independent of the measurement model, the BIC performs poorly in the static environment, while the entropy-based approach performs poorly in the changing environment. The proposed probabilistic approach performs well in all simulated situations. In particular, it produces very few false positives and therefore shows a performance close to the ground truth in the static environment.

B. Real-world experiments

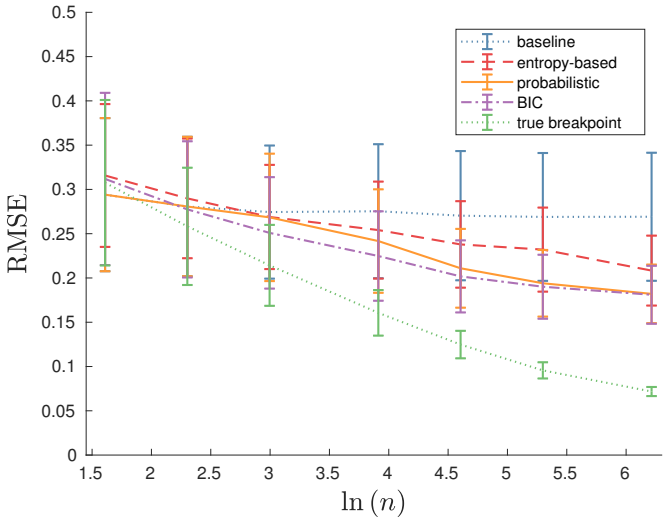
To test our approach in the real world, we create a dataset in a hall with a static Velodyne HDL-64E lidar sensor. We record the scene 13 times with an average of 65 scans per run. In between the scanning phases, we change the scene by adding and removing objects. Two of the sequences include moving people. Overall, we generate 169 scenarios by concatenating every pair of sequences, including the combinations of each sequence with itself to simulate static scenes. For each scenario, we define the ground truth as the map generated from the second sequence of observations. Fig. 1 shows a typical scene of the overall dataset.

We search for breakpoints between each pair of laser scans. For P_1 in (8), we use the cross-validation values from the simulation runs. We compute the RMSE per voxel between the map generated from TRUE and the maps calculated from the individual approaches. Since the majority of voxels does not change, BASE has a low error. To demonstrate that the individual approaches outperform the baseline-approach, we conduct a one-tailed, paired-sample t-test. The p-values in Tab. I correspond to the probability that the corresponding algorithm produces a smaller error than the baseline approach. The variances of the RMSE over the individual scenarios are always three orders of magnitude smaller than the mean. With the proposed method, all three measures significantly outperform the baseline. For the decay-rate model, the entropy-based approach performs best. Considering that large regions of the environment remain static, the good performance of

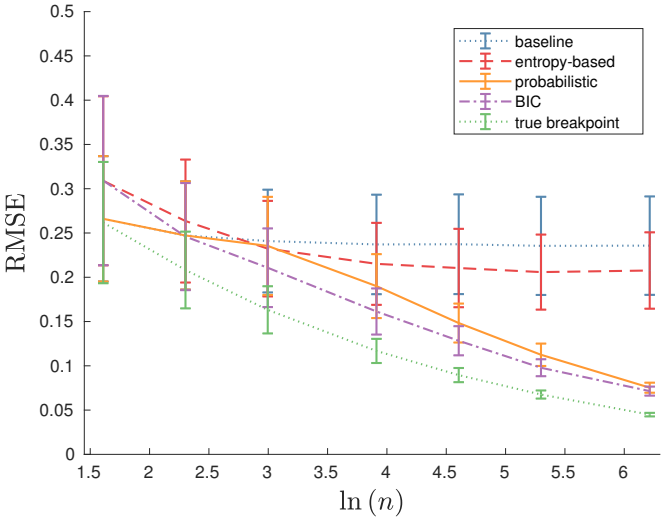
[2]In case of the decay-rate model, the map values λ can vary from zero to infinity. Therefore, we transform λ into the interval $[0, 1]$ by the formula $P_{ref} = 1 - \exp(-\lambda \cdot l)$. Here, P_{ref} can be interpreted as the probability that a perpendicularly incident laser beam is reflected within a voxel with value λ and edge length l.



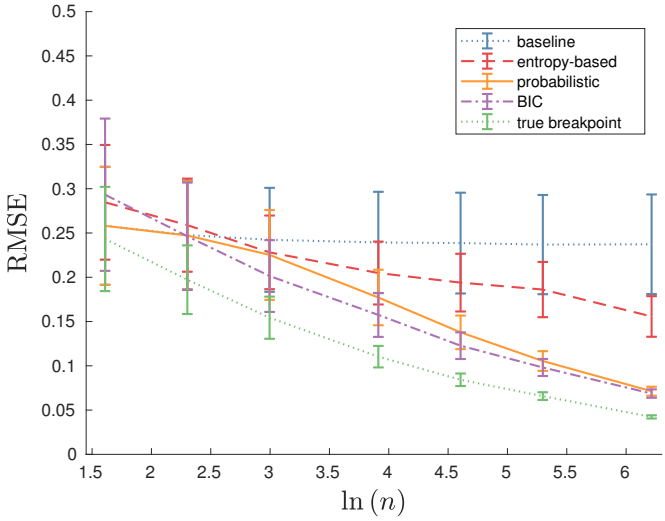
(a) Reflection model in a environment with multiple changes.



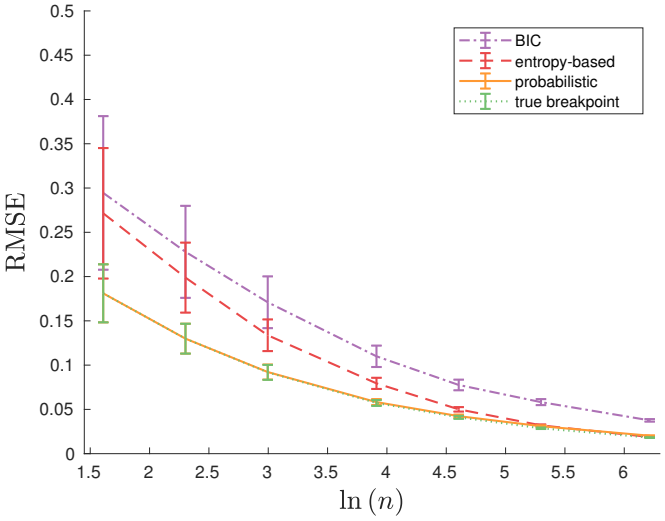
(b) Decay-rate model in a environment with multiple changes.



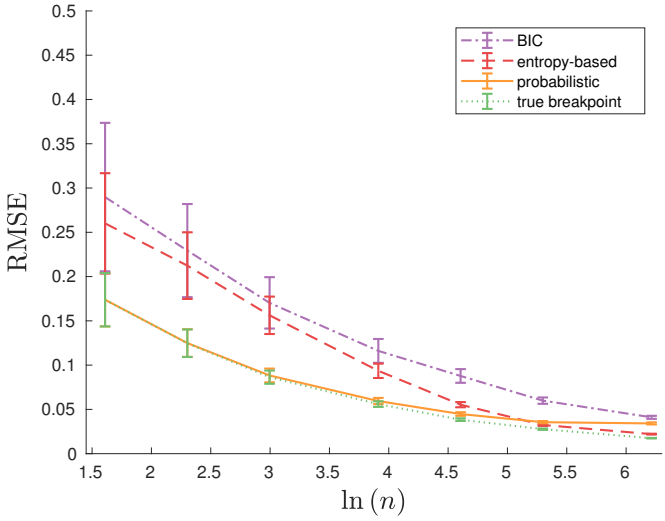
(c) Reflection model in a changing environment.



(d) Decay-rate model in a changing environment.



(e) Reflection model in a static environment.



(f) Decay-rate model in a static environment.

Fig. 2: Mapping error (RMSE) in a simulated environment. The parameter n is the number of observations per voxel during the mapping process. The error bars represent the variances over 10,000 voxels. Note that in the static scenario, the baseline approach and the approach based on the true breakpoint coincide.

Reflection model						
$l[cm]$	BIC	ENT	PRO	BIN	NDT	BASE
10	$\mathbf{0.42(100)}$	$0.64(100)$	$0.45(100)$	$0.74(12.35)$	$1.17(0)$	0.70
30	$3.92(100)$	$4.35(100)$	$\mathbf{3.86(100)}$	$3.91(100)$	$11.08(0)$	4.55
50	$9.44(100)$	$10.25(99.43)$	$9.94(99.95)$	$\mathbf{8.39(100)}$	$28.91(0)$	10.32
Decay-rate model						
$l[cm]$	BIC	ENT	PRO	BIN	NDT	BASE
10	$1.00(100)$	$\mathbf{0.87(100)}$	$0.93(100)$	$1.33(63.55)$	$2.77(0)$	1.35
30	$6.53(99.99)$	$\mathbf{4.59(100)}$	$5.52(100)$	$5.97(100)$	$15.40(0)$	7.15
50	$13.03(100)$	$\mathbf{7.75(100)}$	$10.73(100)$	$11.65(99.97)$	$35.28(0)$	14.16

TABLE I: RMSE per voxel of estimated map values for the different algorithms in the real-world experiments for different voxel edge lengths. We show its mean in $[10^{-4}m]$ over the individual scenarios and omit the corresponding variances as they are always at least three orders of magnitude smaller. In the brackets, we show the p-values of a paired sample t-test between the corresponding algorithms and the baseline. The p-values are in percent and rounded within precision of 10^{-4}.

ENT is in accordance with the simulation results. For the reference model in NDT, we use all measurements before the breakpoint. Since NDT only considers the end points of the laser beams, it cannot detect the removal of objects. In our datasets, it performs worse than the baseline approach as it assumes that the spatial distribution of points within different laser scans remains constant in a static scene. This does not hold for our experiments, resulting in false positives.

V. Conclusion

We propose a method to detect changes in environments represented by real-valued grid maps generated from lidar measurements. Our approach uses different measures to score each point in time according to the probability that it constitutes a change in the environment. The score depends on the two map posteriors: the one given all measurements before this point in time and the one given all measurements after. We compare three baseline approaches and three variants of our method based on different measures in simulated and real-world experiments. While our method always outperforms the baseline in these experiments, we plan to investigate how the proposed measures perform in different scenarios. One advantage of our method is that it works on the basis of voxels and does not require any semantic scene understanding. However, in the future, we plan to relax the assumption of independent voxels to detect changes on the level of whole objects instead. As presented in this paper, the proposed algorithm serves as method for mapping with known poses. In the future, we plan to embed it into a SLAM framework.

References

[1] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, June 1989.

[2] ——, "Occupancy grids: a probabilistic framework for robot perception and navigation," Ph.D. dissertation, Carnegie Mellon University, 1989.

[3] H. Moravec, "Sensor fusion in certainty grids for mobile robots," *AI Mag.*, vol. 9, no. 2, pp. 61–74, July 1988.

[4] D. Hähnel, R. Triebel, W. Burgard, and S. Thrun, "Map building with mobile robots in dynamic environments," in *2003 IEEE International Conference on Robotics and Automation. Proceedings*, 2003.

[5] A. Schaefer, L. Luft, and W. Burgard, "An analytical lidar sensor model based on ray path information," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1405–1412, 2017.

[6] L. Luft, A. Schaefer, T. Schubert, and W. Burgard, "Closed-form full map posteriors for robot localization with lidar sensors," in *IEEE International Conference on Intelligent Robots and Systems*, 2017.

[7] D. Arbuckle, A. Howard, and M. Mataric, "Temporal occupancy grids: a method for classifying the spatio-temporal properties of the environment," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, 2002, pp. 409–414 vol.1.

[8] D. Fox, W. Burgard, and S. Thrun, "Markov localization for mobile robots in dynamic environments," *Journal of Artificial Intelligence Research*, vol. 11, pp. 391–427, 1999.

[9] M. Luber, G. Diego Tipaldi, and K. O. Arras, "Place-dependent people tracking," *Int. J. Rob. Res.*, vol. 30, no. 3, Mar. 2011.

[10] J. Saarinen, H. Andreasson, and A. J. Lilienthal, "Independent markov chain occupancy grid maps for representation of dynamic environment," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2012, pp. 3489–3495.

[11] D. Meyer-Delius, M. Beinhofer, and W. Burgard, "Occupancy grid models for robot mapping in changing environments," in *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012, pp. 2024–2030.

[12] P. Biber and T. Duckett, "Dynamic maps for long-term operation of mobile service robots," in *Proceedings of Robotics: Science and Systems*, Cambridge, USA, 2005.

[13] B. Yamauchi and R. Beer, "Spatial learning for navigation in dynamic environments," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 26, no. 3, pp. 496–505, Jun 1996.

[14] M. Fehr, F. Furrer, I. Dryanovski, J. Sturm, I. Gilitschenski, R. Siegwart, and C. Cadena, "TSDF-based change detection for consistent long-term dense reconstruction and dynamic object discovery," in *IEEE International Conference on Robotics and Automation*, 2017, pp. 5237–5244.

[15] T. Krajnik, J. P. Fentanes, G. Cielniak, C. Dondrup, and T. Duckett, "Spectral analysis for long-term robotic mapping," in *IEEE International Conference on Robotics and Automation*, 2014, pp. 3706–3711.

[16] H. Andreasson, M. Magnusson, and A. Lilienthal, "Has something changed here? autonomous difference detection for security patrol robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, pp. 3429–3435.

[17] J. Andrade-Cetto and A. Sanfeliu, "Concurrent map building and localization on indoor dynamic environments," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 16, no. 03, pp. 361–374, 2002.

[18] B. Sofman, B. Neuman, A. Stentz, and J. A. Bagnell, "Anytime online novelty and change detection for mobile robots," *Journal of Field Robotics*, vol. 28, no. 4.

[19] L. J. Manso, P. Nez, S. da Silva, and P. Drews-Jr, "A novel robust scene change detection algorithm for autonomous robots using mixtures of gaussians," *International Journal of Advanced Robotic Systems*, vol. 11, no. 2, p. 18, 2014.

[20] G. Schwarz, "Estimating the dimension of a model," *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.