

Confidence-rich Grid Mapping

Ali-akbar Agha-mohammadi, Eric Heiden, Karol Hausman, Gaurav Sukhatme

Abstract Occupancy grids are common frameworks for creating a map of the environment using a robot. Traditional occupancy grid mapping algorithms assume that map voxels are independent of each other. In addition, they utilize a map representation, wherein each voxel stores a single number representing the occupancy probability. This leads to conflicts in the map, and more importantly, inconsistency between the map error and the reported confidence values. Such inconsistencies pose challenges for planners that rely on the generated map for collision avoidance. This paper studies occupancy grids from a planning perspective and proposes a new grid mapping algorithm in the presence of noisy measurements. In our novel representation of the map, we store richer data at each voxel, which includes an accurate estimate of the variance of occupancy. We show that in addition to achieving maps that are more accurate than traditional methods, the proposed filtering scheme demonstrates a much higher level of consistency between its error and the reported confidence.

1 Introduction

Consider a quadrotor equipped with a forward-facing stereo camera flying in an obstacle-laden environment tasked to reach a goal. In order to ensure the safety of the system and avoid collisions, the robot needs to create a representation of obstacles, which we refer to as the map, and incorporate it in the planning framework. Due to the noise caused by using imperfect sensors and models, the robot requires a probabilistic representation of the map that is able to capture the uncertainty of the environment. In order to plan trajectories using such a probabilistic map, the planner needs to be aware of not only the occupancy value estimates but also of how much these values can be trusted. In particular, this trust (or confidence) is important when considering sensors with high noise, such as stereo cameras with small baseline (see

Ali-akbar Agha-mohammadi
Jet Propulsion Laboratory, Caltech, Pasadena, CA, e-mail: aliagha@jpl.nasa.gov

Eric Heiden, Karol Hausman, Gaurav Sukhatme
University of Southern California e-mail: heiden@usc.edu

Fig. 1). This paper presents an algorithm that creates a map which in addition to the most likely occupancy values encodes the confidence (trust) associated with these values.

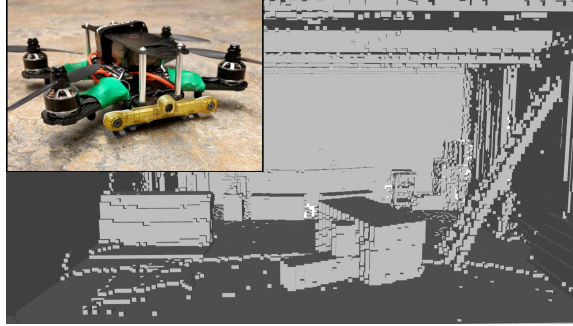


Fig. 1 Top left: example of a micro aerial vehicle capable of fast and dynamic flight up to 20mph. The vehicle is equipped with stereo cameras (baseline 80mm) that are used as the main depth sensor. Background: occupancy grid map of Vicon Room 1 from the EuRoC MAV Dataset [1].

Grid-based structures are among the most common representations of the environment when dealing with range sensors [26]. Typically, each grid voxel contains binary information indicating whether the voxel is free or occupied. In a slightly richer format, each voxel contains the probability of being occupied. In the main body of literature occupancy grids are used to store binary occupancies updated by the log-odds method [22, 23], which is discussed in detail in Sec. 3. Even though the log-odds-based occupancy grids have enjoyed success in a variety of applications, these methods, especially when coping with noisy sensors such as sonar and stereo cameras, suffer from three main issues:

- A) The occupancy of each voxel is updated independently of the rest of the map. This is a well-known problem [22] which has been shown to lead to conflicts between map and measurement data. In particular, when the sensor is noisy or has a large field of view there is a clear coupling between voxels that fall into the field of view of the sensor.
- B) The log-odds methods rely on the “inverse sensor model” (ISM), which needs to be hand-engineered for each sensor and a given environment.
- C) In order to represent the voxel occupancy, each voxel stores a single number. As a result, there is no consistent confidence or trust value to help the planner to decide the reliability of the estimated occupancy.

In this paper, we propose a method that partially relaxes these assumptions, generates more accurate maps, and provided a more consistent filtering mechanism than prior approaches. The highlights and contributions of this work are as follows:

1. The main assumption in traditional occupancy grid mapping is (partially) relaxed. We take into account the dependence between voxels in the measurement cone at every step. Further, the proposed method relaxes the binary assumption on the occupancy level and it is capable of coping with maps where each voxel is only partially occupied by obstacles.

2. We replace the ad-hoc inverse sensor model by a novel “sensor cause model”, which is computed based on the forward sensor model in a principled manner.
3. In addition to the most likely occupancy value for each voxel, the proposed map representation contains confidence values (e.g. variance) of voxel occupancies. The confidence information is crucial for planning over grid maps. We incorporate the sensor model and its uncertainties in characterizing the map accuracy.
4. While the majority of approaches that relax the voxel-independence assumption are batch methods, our method does not require logging of the data in an offline phase. Instead, the map can be updated online as the sensory data are received.

In our experiments, our method achieves an improvement over traditional approaches of up to 30% in absolute error and up to two orders of magnitudes better variance consistency, according to the proposed inconsistency measure. We believe that the map representation proposed here and the probabilistic map-update algorithm provide a significant step towards uncertainty-aware safe planning, a crucial component to enabling fast navigation in uncertain environments.

2 Related Work

The first application of occupancy grids in robotics dates back to Moravec [12] and Elfes [2] and has since been widely used in robotics. Thrun et al. [22], Stachniss [19], and Thrun et al. [23] discuss many variants of these methods.

Grid-based maps have been constructed using a variety of ranging sensors, that include stereo-cameras [10], sonars [27], laser range finders [20], and fusion of thereof [12]. Their structure has been extended to achieve more memory-efficient maps [26]. There have also been methods that extend grid-based mapping to store richer forms of data, including distance to obstacle surface [14], reflective properties of environment [4], and color/texturiness [13]. The main method used to updating the occupancy values of the voxels was presented by Thrun et al. [23] and is based on the log-odds approach. We provide a detailed description of this method in Sec. 3. Another class of mapping methods that have shown great success is the class of Gaussian Process-based mapping methods (e.g., [7, 8, 15, 16, 18, 25]). These methods do not do not rely on voxel grids but model the occupied spaces continuously. Furthermore, they take into account the spatial correlations between occupancy of different regions of the map.

Different researchers have studied the drawbacks of the log-odds approach in occupancy grids and proposed methods to alleviate them [3, 9, 11, 17, 21, 24]. All these methods attempt to alleviate the negative effects caused by the incorrect voxel-independence assumption in mapping. In particular, Thrun [21] proposes a grid-mapping method using forward sensor models, which takes into account all voxel dependencies and achieves maps with higher quality compared to maps resulted from ISM. However, this method requires the measurement data to be collected offline. Then, it runs the expectation-maximization (EM) algorithm on the full data to compute the most likely map. Hahnel et al. [3] extends the grid-based mapping methods to dynamic environments using a similar sensor model to the one used in this paper. However, the method assumes accurate measurements (e.g. coming from a laser range finder). It also uses EM to compute the map, which limits the result to the most likely values and does not provide any confidence measure on the reported

values. In this paper, we propose a mapping method that is online and can cope with high-noise range measurements by incorporating the noise in the model. More importantly, the proposed method additionally computes a confidence value for the estimate which can be very beneficial for planning purposes.

3 Occupancy grid mapping using inverse sensor models

Most occupancy grid mapping methods decompose the full mapping problem to many binary estimation problems on individual voxels assuming full independence between voxels. This assumption leads to inconsistencies in the resulting map. We discuss the method and these assumptions in this section.

Let $G = [G^1, \dots, G^n]$ be an n -voxel grid overlaid on the 3D (or 2D) environment, where $G^i \in \mathbb{R}^3$ is a 3D point representing the center of the i -th voxel of the grid in the global coordinate frame. An occupancy map $m = [m^1, \dots, m^n]$ is defined as a set of values over this grid. We start with a more general definition of occupancy where $m^i \in [0, 1]$ denotes what percentage of a voxel is occupied. $m^i = 1$ when the i -th voxel is fully occupied and $m^i = 0$ when it is free. For maps where occupancy can only be 0 or 1, we use the notation $^{bin}m^i$ for the occupancy of voxel i to explicitly show that the occupancy is binary, i.e., $^{bin}m^i \in \{0, 1\}$.

The full mapping problem is defined as estimating map m based on obtained measurements and robot poses. We denote the sensor measurement at the k -th time step by z_k and the sensor configuration at the k -th time step by x_k . Formulating the problem in a Bayesian framework, we compress the information obtained from past measurements $z_{0:k} = \{z_0, \dots, z_k\}$ and $x_{0:k} = \{x_0, \dots, x_k\}$ to create a probability distribution (belief) \bar{b}_k^m on the map m .

$$\bar{b}_k^m = p(m|z_{0:k}, x_{0:k}) \quad (1)$$

However, due to challenges in storing and updating such a high-dimensional belief, grid mapping methods start from individual cells (marginal distributions).

Assumption 1. Collection of marginals: Map pdf is represented by the collection of individual voxel pdfs (marginal pdfs), instead of the full joint pdf.

$$b_k^m \equiv (b_k^{m^i})_{i=1}^n, \quad b_k^{m^i} = p(m^i|z_{0:k}, x_{0:k}) \quad (2)$$

where n denotes the number of voxels in the map.

To compute the marginal $b_k^{m^i}$ in a recursive manner, the method starts with applying the Bayes rule.

$$b_k^{m^i} = p(m^i|z_{0:k}, x_{0:k}) = \frac{p(z_k|m^i, z_{0:k-1}, x_{0:k})p(m^i|z_{0:k-1}, x_{0:k})}{p(z_k|z_{0:k-1}, x_{0:k})} \quad (3)$$

The main incorrect assumption is applied here:

Assumption 2. (Incorrect) Measurement independence: Standard approaches assume that occupancy of voxels are independent given the measurement history. Mathematically:

$$p(z_k|m^i, z_{0:k-1}, x_{0:k}) \approx p(z_k|m^i, x_k) \quad (4)$$

Remark 1. Note that Assumption 2 (Eq. 4) would be precise if conditioning was over the whole map. In other words,

$$p(z_k|m, z_{0:k-1}, x_{0:k}) = p(z_k|m, x_k) \quad (5)$$

is correct. But, when conditioning on a single voxel, this approximation could be very inaccurate, because a single voxel m^i is not enough to generate the likelihood of observation z . For example, there might even be a wall between m^i and the sensor, and clearly m^i alone cannot tell what range will be measured by the sensor in that case.

When dealing with noisy sensors such as stereo cameras or when dealing with noise-free sensors with large measurement cone (such as sonar) this assumption leads to conflicts in the map and estimation inconsistency.

Inverse sensor model: Following Assumption 2, one can apply Bayes rule to Eq. 4

$$p(z_k|m^i, x_k) = \frac{p(m^i|z_k, x_k)p(z_k|x_k)}{p(m^i|x_k)} \quad (6)$$

which gives rise to the concept of inverse sensor model, i.e., $p(m^i|z_k, x_k)$. The inverse sensor model describes the occupancy probability given a single measurement. The model cannot be derived from a sensor model. However, depending on the application and the utilized sensor, ad-hoc models can be hand-engineered. The reason to create this model is that it leads to an elegant mapping scheme on binary maps as follows.

Plugging (4) and (6) into (3), we get:

$$b_k^{m^i} = p(m^i|z_{0:k}, x_{0:k}) = \frac{p(m^i|z_k, x_k)p(z_k|x_k)p(m^i|z_{0:k-1}, x_{0:k})}{p(m^i|x_k)p(z_k|z_{0:k-1}, x_{0:k})} \quad (7)$$

Given that robot's motion does not affect the map:

$$b_k^{m^i} = p(m^i|z_{0:k}, x_{0:k}) = \frac{p(m^i|z_k, x_k)p(z_k|x_k)p(m^i|z_{0:k-1}, x_{0:k-1})}{p(m^i)p(z_k|z_{0:k-1}, x_{0:k})} \quad (8)$$

Assumption 3. Binary occupancy: To complete the recursion, traditional grid-based methods further assume that the occupancy of voxels are binary, i.e., $^{bin}m^i \in \{0, 1\}$. Thus, $p(^{bin}m^i = 1) = 1 - p(^{bin}m^i = 0)$.

According to Assumption 3, one can define odds r_k^i of occupancy and compute it using Eq.(8):

$$r_k^i := \frac{p(^{bin}m^i = 1|z_{0:k}, x_{0:k})}{p(^{bin}m^i = 0|z_{0:k}, x_{0:k})} = \frac{p(^{bin}m^i = 1|z_k, x_k)p(^{bin}m^i = 0)}{p(^{bin}m^i = 0|z_k, x_k)p(^{bin}m^i = 1)} r_{k-1}^i \quad (9)$$

Remark 2. Relying on Assumption 3 and using odds, removes difficult-to-compute terms from the recursion in Eq. (8).

Further, denoting log-odds as $l_k^i = \log r_k^i$, we can simplify the recursion as:

$$l_k^i = l_{k-1}^i + l_{ISM}^i - l_{prior}^i \quad (10)$$

where, $l_{ISM}^i = \log(p(\text{bin} m^i = 1 | z_k, x_k) p(\text{bin} m^i = 0 | z_k, x_k)^{-1})$ is the log-odds of ISM at voxel i , and $l_{prior}^i = \log(p(\text{bin} m^i = 1) p(\text{bin} m^i = 0)^{-1})$ is the log-odds of prior. ISM is often hand-engineered for a given sensor/environment. Fig. 2 shows the typical form of the ISM function.

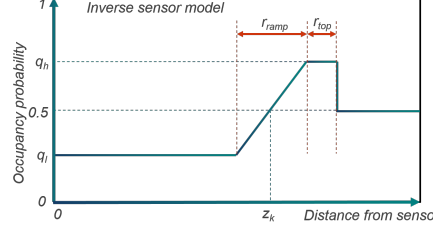


Fig. 2 Typical inverse sensor model for a range sensor. It returns the occupancy probability for voxels on the measurement ray/cone based on their distance to the camera.

4 Confidence-rich Representation

In this paper, we store the probability distribution of m^i in each voxel i . Variable m^i in this paper can be interpreted in two ways:

1. In the more general setting, $m^i \in [0, 1]$ directly represents the occupancy level (the percentage of voxel i that is occupied by obstacles.). The proposed method can model continuous occupancy and Assumption 3 in traditional occupancy mapping can be relaxed.
2. If the underlying true map is assumed to be a binary map (denoted by $\text{bin} m$), the occupancy of the i -th voxel $\text{bin} m^i \in \{0, 1\}$ is distributed as Bernoulli distribution $\text{bin} m^i \sim \text{Bernoulli}(m^i)$. In other words, in this case m^i refers to the parameter of the Bernoulli distribution. While inverse sensor-based mapping methods store m^i as a deterministic value, we estimate m^i probabilistically based on measurements and store its pdf at each voxel. Note that in this setting, $\text{bin} m^i \in \{0, 1\}$, where $m^i \in [0, 1]$ represents the occupancy probability, i.e., $m_k^i = p(\text{bin} m^i = 1 | z_{0:k}, x_{0:k})$.

Problem description: Given the above-mentioned representation, we aim at estimating m based on noisy measurements by computing its posterior distribution. We define three beliefs over the map: (1) full map belief $\bar{b}_k^m = p(m | z_{0:k}, x_{0:k})$, (2) marginal cell beliefs $b_k^m = p(m^i | z_{0:k}, x_{0:k})$, and (3) the collection of marginals $b_k^m = (b_k^m)_{i=1}^M$ ¹. Similar to ISM-based methods, for mapping we maintain and update the collection of marginals b_k^m . To do so, we derive the following items:

1. **Ranging sensor model:** Given the obstacles are described by a stochastic map, we derive a ranging sensor model, i.e., the probability of obtaining measurement z given a stochastic map and robot location: $p(z_k | x_k, b_k^m)$. This model will be utilized in the map update module.

¹ More precisely, in these definitions the variable b refers to the set of parameters that characterize the probability distributions. So, we will treat b as a vector (deterministic or random depending on the context) in the rest of the paper.

2. Recursive density mapping: We derive a recursive mapping scheme τ that updates the current density map based on the last measurements

$$b_{k+1}^{m^i} = \tau^{m^i}(b_k^m, z_{k+1}, x_{k+1}). \quad (11)$$

The fundamental difference with ISM-mapping is that the evolution of the i -th voxel depends on other voxels as well. Note that the input argument to τ^{m^i} is the collection of all voxel beliefs b^m , not just the i -th voxel map b^{m^i} .

3. While planning is beyond the scope of this paper, we briefly discuss how planning can benefit from this enriched map data, and consistent estimation mechanism, to generate actions that actively reduce uncertainty on the map and leads to safer paths.

Overall, this method relaxes Assumptions 2 and 3 of the ISM-based mapping.

5 Range-sensor Modeling

In this section, we model a range sensor when the environment representation is a stochastic map. We focus on passive ranging sensors like stereo cameras, but the discussion can easily be extended to active sensors too.

Ranging pixel: Let us consider an array of ranging sensors (e.g., disparity pixels). We denote the camera center by x_{cam} , the 3D location of the i -th pixel by v , and the ray emanating from x_{cam} and passing through v by $x = (x_{cam}, v)$. Let r denote the distance between the camera center and the closest obstacle to the camera along ray x . In stereo camera range r is related to the measured disparity z as:

$$z = r^{-1} f d_b \quad (12)$$

where, f is camera's focal length and d_b is the baseline between two cameras on the stereo rig. In the following, we focus on a single pixel v and derive the forward sensor model $p(z|x, b^m)$.

Pixel cone: Consider the field of view of pixel v . Precisely speaking, it is a narrow 3D cone with apex at x and boundaries defined by pixel v . Also, for simplicity one can consider just a ray x going through camera center x and the center of pixel v . Pixel cone $\text{Cone}(x)$ refers to the set of voxels in map m that fall into this cone (or lie on ray x). We denote this set by $\mathbb{C} = \text{Cone}(x)$.

Local vs global indices: For a given ray x , we order the voxels along the ray from the closest to the camera to the farthest from the camera. Notation-wise, $i^l \in \{1, \dots, \|\mathbb{C}\|\}$ denote the local index of a voxel on ray x . Function $i^g = g(i^l, x)$ returns the global index i^g of this voxel in the map.

Cause variables: The disparity measurement on pixel v could be the result of light bouncing off any of voxels in the cone $\mathbb{C} = \text{Cone}(x)$ (see Fig. 3). Therefore, any of these voxels are a potential cause for a given measurement. In the case that the environment map is perfectly known, one can pinpoint the exact cause by finding the closest obstacle to the camera center. But, when the knowledge about the environment is partial and probabilistic, the best one can deduce about causes is a probability distribution over all possible causes in the pixel cone $\mathbb{C} = \text{Cone}(x)$.

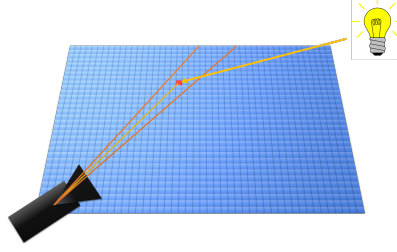


Fig. 3 The cone formed by two red lines depicts the field of view of pixel v . The disparity measurement on pixel v can be caused by light bouncing off any of voxels in the pixel cone and reaching the image plane. In this figure, the measurement is created by light bouncing off the “red” voxel. These causes will play an important role (as hidden variables) in deriving sensor model for stochastic maps.

Cause probability: To derive the full sensor model, we need to reason about which voxel was the cause for a given measurement. For a voxel $c \in \mathbb{C}(x)$ to be the cause, two events need to happen: (i) B^c , which indicates the event of light bouncing off voxel c and (ii) R^c , which indicates the event of light reaching the camera from voxel c .

$$p(c|b^m) = \Pr(B^c, R^c | b^m) = \Pr(R^c | B^c, b^m) \Pr(B^c | b^m) \quad (13)$$

Bouncing probability: To compute the bouncing probability, we rely on the fact that $\Pr(B^c | m^c) = m^c$ (by the definition). Note that $\Pr(B^c | m^c, b^m) = \Pr(B^c | m^c)$.

$$\Pr(B^c | b^m) = \int_0^1 \Pr(B^c | m^c, b^m) p(m^c | b^m) dm^c = \int_0^1 m^c b^{m^c} dm^c = \mathbb{E} m^c = \hat{m}^c$$

Reaching probability: For the ray emanating from voxel c to reach the image plane, it has to go through all voxels on ray x between c and sensor. Let c^l denotes local index of voxel c along the ray x , i.e., $c^l = g^{-1}(c, x)$, then we have:

$$\begin{aligned} \Pr(R^c | B^c, b^m) &= (1 - \Pr(B^{g(c^{l-1}, x)} | b^m)) \Pr(R^{g(c^{l-1}, x)} | B^{g(c^{l-1}, x)}, b^m) \\ &= \prod_{l=1}^{c^l-1} (1 - \Pr(B^{g(l, x)} | b^m)) = \prod_{l=1}^{c^l-1} (1 - \hat{m}^{g(l, x)}) \end{aligned} \quad (14)$$

Sensor model with known cause: Assuming the cause voxel for measurement z is known, the forward sensor is typically modeled as:

$$z = h(x, c, n_z) = \|G^c - x_{cam}\|^{-1} f d_b + n_z, \quad (15)$$

where, $n_z \sim \mathcal{N}(0, V)$ denotes the observation noise, drawn from a zero-mean Gaussian with variance V . We can alternatively describe the observation model in terms of pdfs as follows:

$$p(z|x, c) = \mathcal{N}(\|G^c - x_{cam}\|^{-1} f d_b, V) \quad (16)$$

Sensor model with stochastic maps: Sensor model given a stochastic map can be computed by incorporating hidden cause variables into the formulation:

$$\begin{aligned}
p(z|x; b^m) &= \sum_{c \in \mathbb{C}(x)} p(z|x, c; b^m) p(c|b^m) \\
&= \sum_{c \in \mathbb{C}(x)} \mathcal{N}(\|G^c - x_{cam}\|^{-1} f d_b, V) \hat{m}^c \prod_{l=1}^{c_k^l-1} (1 - \hat{m}^{g(l,x)})
\end{aligned} \tag{17}$$

6 Confidence-Augmented Grid Map

In this section, we derive the recursive mapping algorithm described in Eq. (11). We derive the mapping algorithm τ that can reason not only about the occupancy at each cell, but also about the confidence level of this value. As a result, it enables efficient prediction of the map that can be embedded in planning, resulting in safer plans. We start with a lemma that will be used in derivations.

Lemma 1. *Given the cause, value of the corresponding measurement is irrelevant.*

$$p(m^i|c_k, z_{0:k}, x_{0:k}) = p(m^i|c_k, z_{0:k-1}, x_{0:k})$$

To compute the belief of the i -th voxel, denoted by $b_k^{m^i} = p(m^i|z_{0:k}, x_{0:k})$, we bring the cause variables into formulation.

$$\begin{aligned}
b_k^{m^i} &= p(m^i|z_{0:k}, x_{0:k}) = \sum_{c_k \in \mathbb{C}(x)} p(m^i|c_k, z_{0:k}, x_{0:k}) p(c_k|z_{0:k}, x_{0:k}) \\
&= \sum_{c_k \in \mathbb{C}(x)} p(m^i|c_k, z_{0:k-1}, x_{0:k}) p(c_k|z_{0:k}, x_{0:k}) = \sum_{c_k \in \mathbb{C}(x)} \frac{p(c_k|m^i, z_{0:k-1}, x_{0:k})}{p(c_k|z_{0:k-1}, x_{0:k})} p(c_k|z_{0:k}, x_{0:k}) b_{k-1}^{m^i}
\end{aligned} \tag{18}$$

It can be shown that b_{k-1}^m is sufficient statistics [6] for the data $(z_{0:k-1}, x_{0:k-1})$ in above terms. Thus, we can rewrite (18) as:

$$b_k^{m^i} = \sum_{c_k \in \mathbb{C}(x)} \frac{p(c_k|m^i, b_{k-1}^m, x_k)}{p(c_k|b_{k-1}^m, x_k)} p(c_k|b_{k-1}^m, z_k, x_k) b_{k-1}^{m^i} \tag{19}$$

In the following, we make the assumption that the map pdf is sufficient for computing the bouncing probability from voxel c (i.e., one can ignore voxel i given the rest of the map.) Mathematically, for $c_k \neq i$, we assume:

$$\Pr(B^{c_k}|m^i, b_{k-1}^m, x_k) \cong \Pr(B^{c_k}|b_{k-1}^m, x_k) = \hat{m}^{c_k}$$

Note that we still preserve a strong dependence between voxels via the reaching probability. To see this clearly, let's expand the numerator $p(c_k|m^i, b_{k-1}^m, x_k)$ in (19) as (we drop x to unclutter the equations):

$$\begin{aligned}
p(c_k|m^i, b_{k-1}^m, x_k) &= \Pr(B^{c_k}, R^{c_k}|m^i, b_{k-1}^m, x_k) = \Pr(B^{c_k}|m^i, b_{k-1}^m, x_k) \Pr(R^{c_k}|B^{c_k}, m^i, b_{k-1}^m, x_k) \\
&= \begin{cases} \hat{m}^{c_k} \prod_{l=1}^{c_k^l-1} (1 - \hat{m}^{g(l,x)}) & \text{if } c_k^l < i^l \\ m^i \prod_{l=1}^{c_k^l-1} (1 - \hat{m}^{g(l,x)}) & \text{if } c_k^l = i^l \\ \hat{m}^{c_k} \left(\prod_{l=1}^{i^l-1} (1 - \hat{m}^{g(l,x)}) \right) (1 - m^i) \left(\prod_{l=i^l+1}^{c_k^l-1} (1 - \hat{m}^{g(l,x)}) \right) & \text{if } c_k^l > i^l \end{cases} \tag{20}
\end{aligned}$$

The denominator is $p(c_k|b_{k-1}^m, x_k) = \hat{m}^{c_k} \prod_{l=1}^{c_k^l-1} (1 - \hat{m}^{g(l,x)})$ for all $c_k \in \mathbb{C}(x)$. In these equations, $c_k^l = g^{-1}(c_k, x_k)$ and $i^l = g^{-1}(i, x_k)$ are the corresponding indices of c_k and i in the local frame. Therefore, the ratio in (19) is simplified to:

$$\frac{p(c_k|m^i, b_{k-1}^m, x_k)}{p(c_k|b_{k-1}^m, x_k)} = \begin{cases} 1 & \text{if } c_k^l < i^l \\ m^i (\hat{m}^i)^{-1} & \text{if } c_k^l = i^l \\ (1 - m^i)(1 - \hat{m}^i)^{-1} & \text{if } c_k^l > i^l \end{cases}$$

Plugging the ratio back into the (19), and collecting linear and constant terms, we can show that:

$$p(m^i|z_{0:k}, x_{0:k}) = (\alpha^i m^i + \beta^i) p(m^i|z_{0:k-1}, x_{0:k-1}) \quad (21)$$

where

$$\alpha^i = \sum_{c_k^l=1}^{i^l-1} p(c_k|b_{k-1}^m, z_k, x_k) + (1 - \hat{m}^i)^{-1} \sum_{c_k^l=i^l+1}^{|\mathbb{C}(x)|} p(c_k|b_{k-1}^m, z_k, x_k) \quad (22)$$

$$\beta^i = (\hat{m}^i)^{-1} p(c_k|b_{k-1}^m, z_k, x_k) - (1 - \hat{m}^i)^{-1} \sum_{c_k^l=i^l+1}^{|\mathbb{C}(x)|} p(c_k|b_{k-1}^m, z_k, x_k) \quad (23)$$

In a more compact form, we can rewrite Eq. (21) as:

$$b_{k+1}^{m^i} = \tau^i(b_k^m, z_{k+1}, x_{k+1}). \quad (24)$$

Sensor cause model: The proposed machinery gives rise to the term $p(c_k|z_{0:k}, x_{0:k}) = p(c_k|b_{k-1}^m, z_k, x_k)$, which is referred to as ‘‘Sensor Cause Model (SCM)’’ in this paper. As opposed to the inverse sensor model in traditional mapping that needs to be hand-engineered, the SCM can be derived from the forward sensor model in a principled way as follows. η' is the normalization constant.

$$\begin{aligned} p(c_k|z_{0:k}, x_{0:k}) &= p(c_k|b_{k-1}^m, z_k, x_k) = \frac{p(z_k|c_k, x_k) p(c_k|b_{k-1}^m, x_k)}{p(z_k|b_{k-1}^m, x_k)} \\ &= \eta' p(z_k|c_k, x_k) p(c_k|b_{k-1}^m, x_k) = \eta' p(z_k|c_k, x_k) \hat{m}_{k-1}^{c_k} \prod_{j=1}^{c_k^l-1} (1 - \hat{m}_{k-1}^{g(j,x)}), \forall c_k \in \mathbb{C}(x_k) \end{aligned} \quad (25)$$

6.1 Algorithm and Computational Complexity

The complete belief-update algorithm is presented in Alg. 1. For every measurement obtained from a single ray, we find all the voxels \mathbf{V}_{ray} that the ray traverses (line 1). In the next step, the sensor cause model is computed (line 2). It is worth noting that, if implemented recursively, this step has linear complexity w.r.t. the number of voxels on the ray, i.e., $\mathcal{O}(|\mathbf{V}_{\text{ray}}|)$. The terms in Eqs. (22) and (23) have been computed in Line 2 and stored as SCM. In lines 4 and 5, we retrieve this values and compute the likelihood parameters α , β and concurrently update the voxel belief.

The voxel belief can be represented using a standard particle filter with a constant number of particles. Doing these steps at the same time enables to use recursion again, which results in a linear complexity w.r.t. the number of voxels on the ray, i.e. $\mathcal{O}(|\mathbf{V}_{\text{ray}}|)$.

We assume a KD-tree to be used as the data structure for storing the map, in which we can retrieve the voxel of interest in $\mathcal{O}(\log(|\mathbf{V}|))$ time, where \mathbf{V} is the set of voxels in the map (line 1). The resulting complexity of the proposed algorithm is $\mathcal{O}(|\mathbf{V}_{\text{ray}}| \cdot \log(|\mathbf{V}|) \cdot |\mathbf{V}_{\text{ray}}|) = \mathcal{O}(|\mathbf{V}_{\text{ray}}|^2 \cdot \log(|\mathbf{V}|))$. Assuming the same data structure for storing the map, the standard log-odds approach results in the complexity $\mathcal{O}(|\mathbf{V}_{\text{ray}}| \cdot \log(|\mathbf{V}|))$.

It is important to note that the increase in complexity is minimal for the proposed approach. In addition, the complexity of our method is still independent of the size of the map $|\mathbf{V}|$. This way, we increase the amount of information stored in the map without substantially increasing the complexity of the mapping algorithm.

Algorithm 1: Confidence-rich grid mapping

input : Current map belief b_k , Current observation z_k , current robot measurement ray x_k
output : Updated map belief b_{k+1}
Procedure : $b_{k+1} = \text{Update}(b_k, z_k, x_k)$
1 Find \mathbf{V}_{ray} voxels on the ray x_k ;
2 Compute SCM using Eq. (25);
3 **foreach** $v_i \in \mathbf{V}_{\text{ray}}$ **do**
4 Compute α^i, β^i (Eqs. (22), (23));
5 Update voxel belief (Eq. (21));
6 **return** b_{k+1} ;

7 Confidence in Map and Safe Exploration for Planning

The proposed method not only provides a more accurate map estimate, but more importantly, the uncertainty associated with the returned value. In doing so, it incorporates the full forward sensor model into the mapping process. As an example, it can distinguish between two voxels, when they are both reported as almost free (e.g. $\hat{m}^1 = \hat{m}^2 = 0.1$), but they have different confidence levels (e.g. $\sigma^{m^1} = 0.01$ and $\sigma^{m^2} = 0.2$). This confidence level is a crucial piece of information for the planner. Obviously, the planner should either try to avoid m^2 since the robot is not sure if m^2 is actually risk free (due to high variance), or it needs to take active perceptual actions to take more measurements from m^2 before passing through that part of the map.

In the log-odds-based method only one number is stored in the map, namely the parameter of the Bernoulli distribution. One might try to utilize the variance of the Bernoulli distribution to infer the confidence in an log-odds-based map, but due to the incorrect assumptions made in the mapping process and since the Bernoulli distribution is a single parameter distribution (mean and variance are dependent), the computed variance is not a reliable confidence source.

It is important to note that generally a planner is able to cope with large errors “if” there is a high variance associated with them. However, if the error is high and the filter is confident about its wrong estimate, planning becomes very risky, and prone to fail.

Consider a simple planning scenario in an unknown environment, where a quadrotor has to traverse the environment as fast as possible while ensuring a reasonable level of safety. Using our confidence-rich mapping (CRM) approach, one can predict the future variance of the occupancy of the path. Since our method yields consistent variance estimates, it enables uncertainty-aware planner to reliably infer about the information gain of future perceptual measurements. Having this information opens new possibilities for uncertainty-aware planners such as incorporating safety-critical exploratory behavior into the fast navigation task. We plan to thoroughly investigate the planning component of the problem in future work.

8 Results

In this section, we demonstrate the performance of the proposed method and compare it with commonly used mapping methods. We start by studying the mapping error and then we discuss the consistency of the estimation process.

8.1 Experimental Setup

Fig. 4 shows a 2D ground truth map which serves as a simulation environment for the following scenario. Each voxel is assumed to be a cube with $s = 0.06\text{m}$ side length. The environment size is $2 \times 2\text{m}$, consisting of 1089 voxels. Each voxel is either fully occupied (shown in black) or empty (white). The robot follows a trajectory (red arrows), as shown in Fig. 4, and takes one measurement at every arrow position.

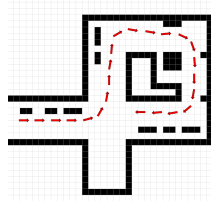


Fig. 4 Ground truth map with sampling trajectory where red arrows represent measurement positions.

sensor std dev	$0.25s$	$0.5s$	s	$2s$	$3s$
log-odds MAE	0.346	0.348	0.351	0.358	0.376
GPOM MAE	0.421	0.468	0.468	0.468	0.468
CRM MAE	0.289	0.299	0.323	0.355	0.365
log-odds I_c	17.340	18.609	17.665	18.862	19.243
GPOM I_c	10.167	10.172	12.964	10.212	10.490
CRM I_c	0.056	2.040	6.100	5.418	4.575

Table 1 Mean Absolute Error (MAE) and inconsistency I_c with $\gamma = 1$ (Eq. 27) of log-odds mapping, GPOM and the proposed method under different sensor model noise std. deviations (right) where s is the voxel size (0.06m).

For the sensing system, we simulate a ranging sensor with 60 omnidirectional depth sensors spanning over a field-of-view of 360° and reaching up to a range of 1m. Measurements are corrupted by zero-mean Gaussian noise with 0.06m std. deviation, i.e. one voxel length.

For the log-odds mapping, we use a typical inverse sensor model (Fig. 2) with parameters $r_{ramp} = 0.1$, $r_{top} = 0.1$, $q_l = 0.45$, and $q_h = 0.55$. The Gaussian Processes occupancy mapping (GPOM) uses a combination of a Matérn kernel with a Gaussian noise:

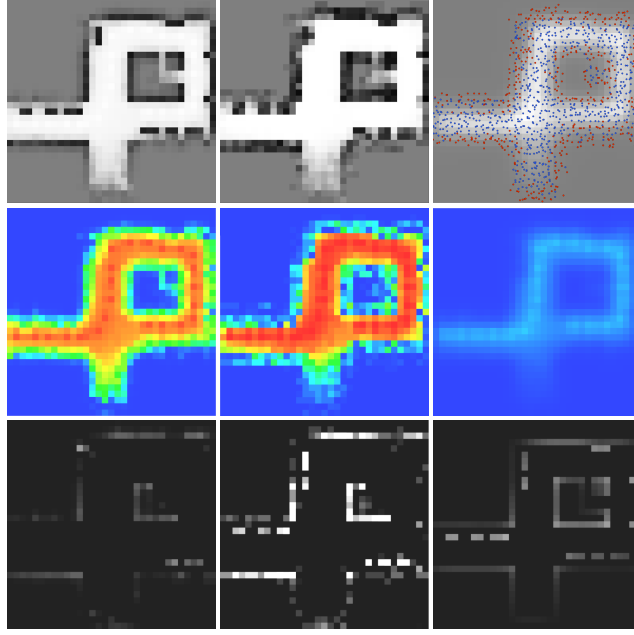


Fig. 5 Mapping results: CRM (left column), Log Odds (center column), GPOM (right column). Mean occupancies (top row), estimated standard deviation (middle row), inconsistencies $|e_c| > \sigma_c$ for absolute voxel error $|e_c|$ and std σ_c (lower row). The mean occupancy GP map (top right) shows the GP sample set where blue dots represent samples of free space and red dots represent occupied space. Measurement rays are discretized on the voxel grid for GPOM, and appear with a random offset only for visualization purposes.

$$k(x, x') = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}|x-x'|}{l} \right)^\nu K_\nu \left(\frac{\sqrt{2\nu}|x-x'|}{l} \right) + \sigma^2 \delta_{x,x'} \quad (26)$$

where K_ν is the modified Bessel function of order ν and $\delta_{x,x'}$ is the Kronecker delta. This kernel has three parameters, ν, l and σ , which have been determined empirically (Fig. 9(b)) to achieve comparable results to log-odds mapping and our method.

The occupancy maps resulting from the log-odds mapping, GPOM and the proposed method are shown in the upper row of Fig. 5, respectively.

8.2 Simulation Experiments

First, we study the sensitivity of our method to different sensor noises. The top rows of Table 1 show the map mean absolute error (MAE) over time for different sensor noise std deviations. The MAE is averaged over all the voxels that were updated throughout the mapping process to best show the improvement of the affected parts of the map. For the same noise intensity, the proposed CRM method shows a smaller error compared to the log-odds and Gaussian Processes method.

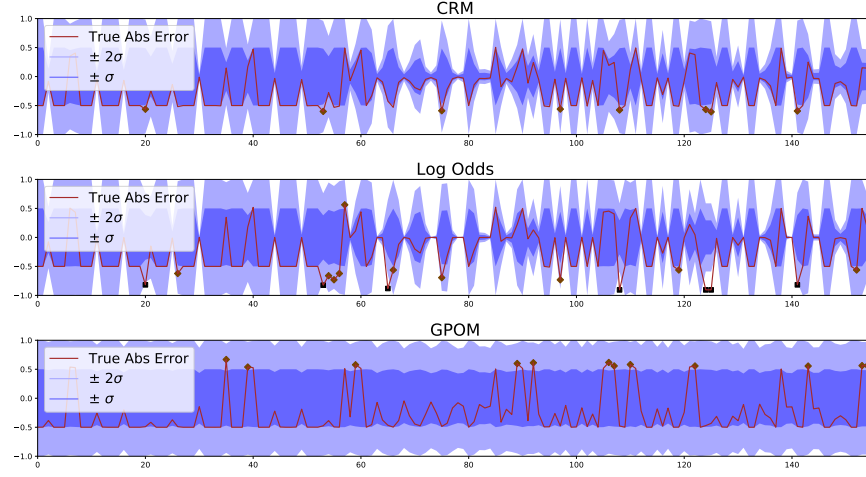


Fig. 6 Mapping error (red) and algorithmically computed std deviation (blue shades for 2σ (light) and σ (dark) confidence bounds) over a set of voxels, with highlighted severe inconsistencies: errors outside 2σ -interval (black squares) and outside 1σ -interval (orange diamonds).

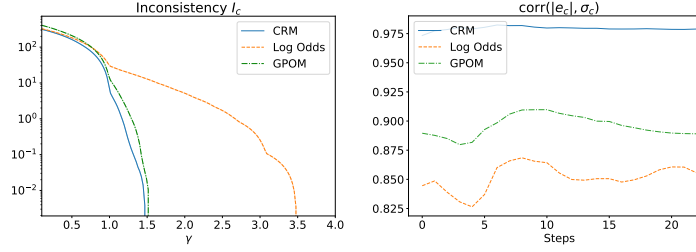


Fig. 7 Evaluation of inconsistency (Eq. 27) over threshold values γ (left). Pearson correlation coefficient between the reported std deviation and the absolute error over all voxels at a given step (right).

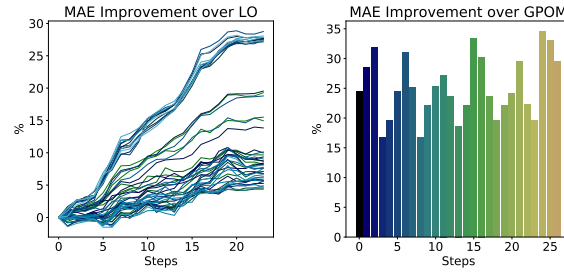


Fig. 8 Percentage improvement in mean absolute error (MAE) of CRM over Log Odds with different ISM parameters (left) and Gaussian Processes with different GP parameters (right).

It is worth emphasizing that reducing the map error is only an ancillary benefit of our method. The main objective of CRM is to provide a consistent confidence measure. This consistency is particularly important for planning purposes. A planner might be able to handle large errors as long as the filter indicates that the estimates are unreliable (e.g. via their variance). However, if the filter returns a wrong estimate with low variance (i.e. it is confident that its estimate is correct, while it is not), then the planner is prone to yield unsafe results.

To quantify the inconsistency between the error and reported variances, we utilize the following measure:

$$I_c = \sum_c \text{ramp}(|e_c| - \gamma\sigma_c) \quad (27)$$

where, e_c and σ_c denote the estimation error and the std. deviation of voxel c , respectively. γ decides what level of error is acceptable. The ramp function $\text{ramp}(x) := \max(0, x)$ ensures that only inconsistent voxels (with respect to $\gamma\sigma$) contribute to the summation. Accordingly, I_c indicates how much of the error signal is out of bound (i.e., how unreliable the estimate is) over the whole map. As can be seen in the middle row of Fig. 5, where voxels of high inconsistency appear brighter, the log-odds based approach tends to be overly confident in false estimates, compared to the other methods. GP-based maps appear less clearer compared to traditional occupancy grids with the advantage of a higher consistency. However, as can be seen in the bottom right map of Fig. 5, smaller obstacles are not mapped accurately and pose dangerous areas of inconsistency.

The last two rows of Table 1 show the described inconsistency measure for different sensor noise levels. The proposed CRM method is able to significantly outperform the log-odds and the Gaussian Processes methods for all of the tested noise std. deviations.

Fig. 6 shows the error value corresponding to the groundtruth map in Fig. 4 along with the estimated σ -bound over $1/7$ of all voxels. As can be seen from the top row, the σ -bound from CRM grows and shrinks in a consistent manner with the error, and behaves as a reliable confidence interval that can be used in planning. However, the log-odds mapping (middle row), there exist many voxels (highlighted by square and diamond markers) where the error is high (close to -1 or 1, i.e., a free voxel is estimated as fully occupied or vice versa) *at which the variance is very low*, which pose a significant challenge to the planner. Similarly to our method, GPOM rarely exhibits inconsistencies. However, the σ -bound is not as pronounced (bottom row), compared to CRM, making it difficult to assess the confidence of the estimates since most of them appear highly uncertain (σ close to 0.5).

Fig. 7(a) compares the inconsistency measure (Eq. 27) over different γ thresholds for the presented mapping algorithms. The log-odds approach overestimates the confidence significantly more while GPOM and our method yield much safer estimates.

While GPOM and the proposed approach yield similar results in the ramp inconsistency comparison, it should be noted that GPOM is computationally much more expensive than occupancy grids [16]. Gaussian Processes rely on the full set of measurement samples in order to predict the map occupancy. Over the course of the trajectory, it takes more than two minutes to update the Gaussian Processes with

1380 depth measurements and to query the GP at all every voxel. In comparison, on the same hardware, our method finishes both tasks in less than five seconds.

As an alternative measure of consistency, we compute the Pearson correlation coefficient between the error and std. deviation (see Fig. 7(b)), which indicates that the std. deviation generated by the proposed method is highly correlated with the error, hence, it reliably describes the error’s behavior.

Since the inverse sensor model is typically hand-engineered for a given sensor and a given environment, we sweep over a set of inverse sensor models to compare the performance with the proposed method. Following the generic form of ISM in Fig. 2, we create 27 ISM models by (i) setting the $q_h - 0.5 = 0.5 - q_l$ to (0.05, 0.2, 0.4), (ii) setting r_{ramp} to (0.05, 0.1, 0.3), and (iii) setting r_{top} to (0.05, 0.1, 0.3). Accordingly, we draw the GP parameters v , l and σ from the sets $\{2, 2.5, 3\}$, $\{2.5, 3.5, 4.5\}$ and $\{0.5, 1, 1.5\}$, respectively, yielding 27 different GPOM configurations. The results reported in Fig. 9 show that CRM consistently produces a smaller absolute error than log-odds and GPOM over a variety of the ISM and GP parameters, as well as a smaller portion of error that is inconsistent according to the I_c measure.

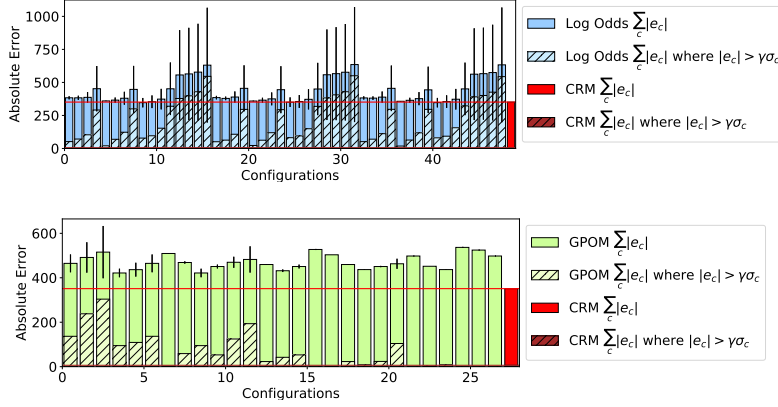


Fig. 9 Comparison of cumulative mapping error between log-odds using different ISM parameters and the proposed method. The whiskers depict the inconsistency I_c (Eq. 27), the striped areas represent the portion of errors $|e_c|$ that are inconsistent with the reported std (greater than $\gamma\sigma_c$ for $\gamma = 1.25$). Our method results in a seven times more consistent mapping with one third of the critical errors due to overconfidence, compared to the best log-odds configuration. While some GPOM configurations yield zero inconsistencies for $\gamma = 1.25$, the absolute error is much higher ($> 20\%$) compared to the proposed method, drawing the overall occupancy information less useful.

8.3 Real Data Experiment

In addition to the extensive simulation experiments, we evaluate the proposed method on the dataset “albert-b-laser-vision” from the Robotics Data Set Repository (Radish) [5]. Thanks go to Cyrill Stachniss for providing this data.

The dataset contains more than 4,000 sampling poses at which 180 laser range measurements were recorded over a field-of-view of 180° (cf. Fig. 10). A groundtruth

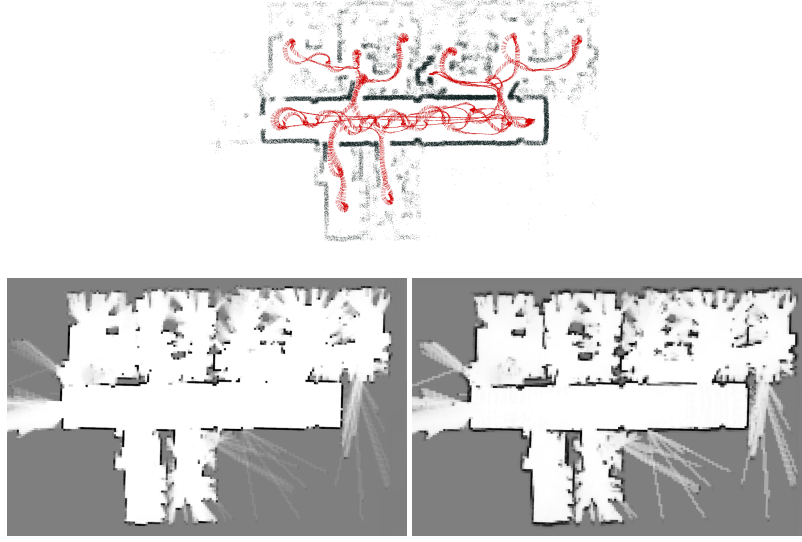


Fig. 10 Top center: end points of laser range measurements (black dots), and robot pose trajectory (red arrows) from the "albert-b-laser-vision" dataset. Bottom left: log-odds mapping results. Bottom right: mapping results from the proposed method.

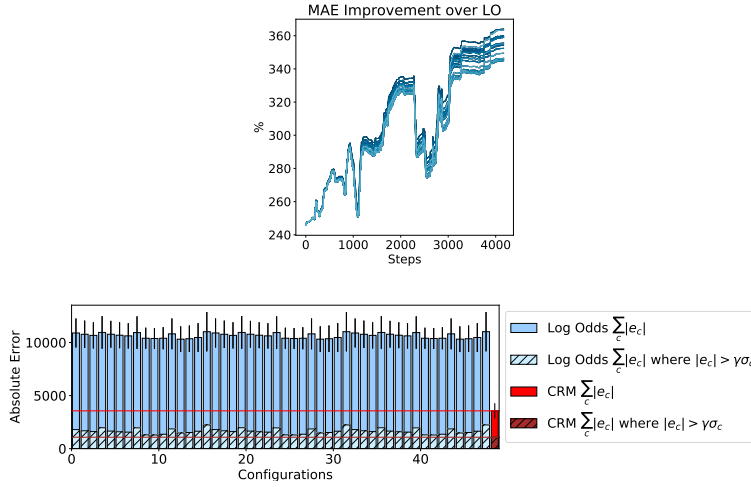


Fig. 11 Top: percentage improvement in mean absolute error (MAE) of CRM over Log Odds with 27 different ISM parameters on the "albert-b-laser-vision" dataset. Bottom: comparison of cumulative mapping error between log-odds with different ISM parameters and our method. The whiskers depict the inconsistency I_c (Eq. 27), the striped areas represent the portion of errors $|e_c|$ that are inconsistent with the reported std (greater than $\gamma\sigma_c$ for $\gamma = 1.25$).

map over a 240×240 grid of voxels with 0.125m side length has been computed using the log-odds mapping method from all of the measurements. In the mapping

evaluation, only every tenth measurement from each sampling (i.e. 10% of the data) has been given as input to log-odds and CRM.

We compared the mapping results from both methods to the groundtruth map, while computing a log-odds mapping over 27 different ISM models, as described in section 8. The proposed methods yielded an improvement in mapping accuracy, i.e. lower mean absolute error (MAE), of more than three times compared to the best performing ISM model (Fig. 11(a)). After all measurements had been processed, the inconsistency (Eq. 27) of the final log-odds mapping was 984.777 compared to an I_c of 716.211 in CRM, while the fraction of erroneous predictions with over-confident variance estimates was 20% higher in the log-odds mapping compared to our method. Over 1000 measurements, the average run-time of log-odds was 1.362s compared to 4.474s of CRM.

9 Conclusion

This paper proposes an alternative algorithm for occupancy grid mapping, by storing richer data in the map. It extends traditional grid mapping in three ways: first, it relaxes the full-independence assumption and incorporates dependence between voxels in the measurement cone into the mapping scheme. Second, it relaxes the need for hand-engineering an inverse sensor model and proposes the sensor cause model that can be derived in a principled manner from the forward sensor model. Third, and most importantly, it provides consistent confidence values over occupancy estimation that can be reliably used in planning. The method runs online as measurements are received and it enables mapping environments where voxels might be partially occupied. The results on real-world laser measurements show that the mapping accuracy is up to three times better than the ISM-based method and, in simulation, 30% better than Gaussian Processes maps. More importantly, according to the proposed consistency measure, the confidence values are up to two orders of magnitude more reliable.

References

- [1] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*, 2016. doi: 10.1177/0278364915620033. URL <http://ijr.sagepub.com/content/early/2016/01/21/0278364915620033.abstract>.
- [2] Alberto Elfes. *Occupancy grids: A probabilistic framework for robot perception and navigation*. PhD thesis, Carnegie Mellon University, 1989.
- [3] Dirk Hahnel, Rudolph Triebel, Wolfram Burgard, and Sebastian Thrun. Map building with mobile robots in dynamic environments. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1557–1563. IEEE, 2003.
- [4] Andrew Howard and Les Kitchen. Generating sonar maps in highly specular environments. In *In Proceedings of the Fourth International Conference on Control Automation Robotics and Vision*, 1996.
- [5] Andrew Howard and Nicholas Roy. The robotics data set repository (radish), 2003. URL <http://radish.sourceforge.net/>.
- [6] Steven M Kay. *Fundamentals of statistical signal processing, volume i: estimation theory*. 1993.

- [7] Soohwan Kim and Jonghyuk Kim. Occupancy mapping and surface reconstruction using local gaussian processes with kinect sensors. *IEEE transactions on cybernetics*, 43(5):1335–1346, 2013.
- [8] Soohwan Kim, Jonghyuk Kim, et al. Recursive bayesian updates for occupancy mapping and surface reconstruction. In *Proceedings of the Australasian Conference on Robotics and Automation*, 2014.
- [9] Kurt Konolige. Improved occupancy grids for map building. *Autonomous Robots*, 4(4):351–367, 1997.
- [10] Kurt Konolige, Motilal Agrawal, Robert C Bolles, Cregg Cowan, Martin Fischler, and Brian Gerkey. Outdoor mapping and navigation using stereo vision. In *Experimental Robotics*, pages 179–190. Springer, 2008.
- [11] Paskin M and Thrun S. Robotic mapping with polygonal random fields. In *Conference on Uncertainty in Artificial Intelligence*, page 450458, 2005.
- [12] Hans P Moravec. Sensor fusion in certainty grids for mobile robots. *AI magazine*, 9(2):61, 1988.
- [13] Hans P. Moravec. Robot spatial perception by stereoscopic vision and 3d evidence grids. *Technical Report CMU-RI-TR-96-34*, Carnegie Mellon University, 1996.
- [14] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011.
- [15] Simon OCallaghan, Fabio T Ramos, and Hugh Durrant-Whyte. Contextual occupancy maps using gaussian processes. In *IEEE International Conference on Robotics and Automation, 2009. ICRA'09*, pages 1054–1060. IEEE, 2009.
- [16] Simon T OCallaghan and Fabio T Ramos. Gaussian process occupancy maps. *The International Journal of Robotics Research*, 31(1):42–62, 2012.
- [17] Daniel Pagac, Eduardo M Nebot, and Hugh Durrant-Whyte. An evidential approach to probabilistic map-building. In *Reasoning with Uncertainty in Robotics*, pages 164–170. Springer, 1996.
- [18] Fabio Ramos and Lionel Ott. Hilbert maps: scalable continuous occupancy mapping with stochastic gradient descent. *The International Journal of Robotics Research*, 35(14):1717–1730, 2016.
- [19] Cyrill Stachniss. *Robotic mapping and exploration*, volume 55. Springer, 2009.
- [20] Sebastian Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998.
- [21] Sebastian Thrun. Learning occupancy grid maps with forward sensor models. *Autonomous robots*, 15(2):111–127, 2003.
- [22] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. MIT Press, Cambridge, MA, 2005.
- [23] Sebastian Thrun et al. Robotic mapping: A survey. *Exploring artificial intelligence in the new millennium*, 1:1–35, 2002.
- [24] Michael Veeck and Wolfram Burgard. Learning polyline maps from range scan data acquired with mobile robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 1065–1070, 2004.
- [25] Jinkun Wang and Brendan Englot. Fast, accurate gaussian process occupancy maps via test-data octrees and nested bayesian fusion. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 1003–1010, 2016.
- [26] Kai M Wurm, Armin Hornung, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. Octomap: A probabilistic, flexible, and compact 3d map representation for robotic systems. In *ICRA 2010 workshop on best practice in 3D perception and modeling for mobile manipulation*, volume 2, 2010.
- [27] Brian Yamauchi. A frontier-based approach for autonomous exploration. In *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 146–151, 1997.