2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Optimizing Online Occupancy Grid Mapping to Capture the Residual Uncertainty

Rehman S. Merali and Timothy D. Barfoot

Abstract— **Occupancy grids have been a popular mapping technique in mobile robotics for nearly 30 years. Occupancy grids offer a discrete representation of the world and seek to determine the occupancy probability of each cell. Traditional occupancy grid mapping methods make two assumptions for computational efficiency and it has been shown that the full posterior is computationally intractable for real-world mapping applications without these assumptions.**

The two assumptions result in tuning parameters that control the information gained from each distance measurement. In this paper, several tuning parameters found in the literature are optimized against the full posterior in 1D. In addition, this paper presents a new parameterization of the update function that outperforms existing methods in terms of capturing residual uncertainty. Capturing the residual uncertainty better estimates the position of obstacles and prevents under- and over-confidence in both the occupied and unoccupied cells. The paper concludes by showing that the new update function better captures the residual uncertainty in each cell when compared to an offline mapping method for realistic 2D simulations.

I. Introduction

Occupancy Grid (OG) mapping is a widely used mapping technique in mobile robots. In contrast to mapping techniques that only seek to locate features or landmarks in the environment, OG mapping discretizes the environment into cells and seeks to determine whether each cell is occupied or unoccupied given, for example, range measurements gathered by a mobile robot. Occupancy grid mapping was first introduced by Moravec and Elfes [18], but improved by Moravec [17] and Elfes [4] by representing the occupancy of each cell by a single binary random variable, then computing the probability of each cell being occupied. However, for computational efficiency, Moravec and Elfes assumed that cells in the occupancy grid are independent of one another and that measurements are conditionally independent. These two assumptions reduce the complexity of OG mapping from exponential to linear in the number of cells through which a measurement passes.

Although the two assumptions offer a significant computation savings, neither is strictly true and they typically lead to an over- or under-confident estimate of the occupancy probability of each cell. This paper will illustrate this over- or under-confidence by comparing traditional OG mapping to the full Bayesian solution, which makes no assumptions. The full Bayesian solution (a.k.a., full solution) will be introduced in Section II and is able to capture the residual uncertainty in the OG map, but is only computationally tractable in 1D.

Occupancy grid maps are often thresholded on a cell-by-cell basis as occupied or unoccupied and then used for path planning, obstacle avoidance, and other tasks [5]. However, accurately computing the residual uncertainty in occupancy grids is useful for applications such as entropy and information gain calculations. These calculations are often used in exploration algorithms to evaluate the expected information gain of scanning from a particular location on the map [12, 22]. Therefore, capturing the residual uncertainty in the map, resulting from noisy sensor measurements, is essential to accurately computing the information in the occupancy grid.

In [15], we proposed a method known as *patch map* that extends the work of Thrun [23] to approximate the full solution. However, both of these methods require either the maximum a posteriori (MAP) estimate of the occupancy grid or the ground truth map (available in simulation) before computing the residual uncertainty in each cell. Thus, in [16], we applied a well-known variant of a Markov Chain Monte Carlo (MCMC) method known as Gibbs sampling [7] to the problem of capturing the residual uncertainty in an occupancy grid map. The paper demonstrates that the MCMC method is quite effective but is not fast enough to run online. Therefore, in this paper we seek to optimize traditional, two-assumption OG mapping to better capture the residual uncertainty in each cell.

Traditional OG mapping incorporates each range measurement using an update function, which we call Δ. Section III will further explain traditional OG mapping and illustrate five different parameterizations of Δ found in the literature as well as introduce a sixth parametrization that better captures the residual uncertainty in the occupancy grid. The tuning parameters of these six parameterizations are then optimized in Section IV by comparing them to the full solution using 1000 1D datasets. The optimized parameters are shown to work in 2D in Section V where they are compared to the aforementioned patch map benchmark solution. Finally, we conclude the paper and offer possible future extensions of this work in Section VI. We believe that this is the first paper to compare the various Δ functions in the literature and optimize them to best capture the residual uncertainty without sacrificing the linear-time update that has made OG mapping a popular choice for online mapping.

This work was supported by a Natural Sciences and Engineering Research Council of Canada Collaborative Research and Development (NSERC CRD) / Canadian Space Agency Partnership Support Program (CSA PSP) grant in cooperation with MDA Space Missions.

R. S. Merali (rehman.merali@utoronto.ca) and T. D. Barfoot (tim.barfoot@utoronto.ca) are with the University of Toronto Institute for Aerospace Studies, 4925 Dufferin St., Toronto, Canada, M3H 5T6.

II. Full Bayesian Solution

Occupancy grid mapping discretizes an analog environment, m, into a regular grid of K cells and then seeks to determine the occupancy of each cell. Each cell in the map is represented by a binary random variable, m_k, where $k = 1 \dots K$, that indicates whether the cell is *occupied*, $m_k = 1$, or *unoccupied*, $m_k = 0$. The true map, known as the ground truth map, correctly indicates the value of each m_k. An OG mapping algorithm seeks to determine the state of each cell in the map, given a set of N range measurements, $z_{1:N}$, and corresponding robot poses, $x_{1:N}$. To simplify the notation, we will not include the subscript $1{:}N$ when referring to the set of all measurements or poses, just as all K cells in the map are referred to as m. OG mapping assumes that the set of robot poses is known. As most range sensors measure the distance to the first occupied cell, f, (and not beyond that) the occupancy of cells inside walls or obstacles cannot be measured. An omniscient ground truth map can accurately represent the state of these cells, but a mapping algorithm cannot. Therefore, it is typical in the literature to refer to these cells as *unknown*.

Range measurements are typically noisy and occupancy grid mapping algorithms use Bayesian methods to compute the probability that each cell is occupied, $p(m_k)$, given a set of noisy range measurements, z, and corresponding robot poses, x. Since the map is divided into K cells, there are $R = 2^K$ possible occupancy grids. We use a superscript to represent these maps: m^r, where $r = 1 \dots R$.

We presented the full solution to OG mapping in [15], but noted that it is computationally expensive to compute for R maps. In this section we show that by constraining the robot's pose, we only need to compute the full solution for K unique maps. By introducing the hidden state of the first occupied cell, $f = 1 \dots F$, we are able to compute the probability of each map given range measurements and corresponding poses, $p(m^r|z, x)$. Throughout this paper, we assume that the range sensor is a narrow-beam sensor such as a laser rangefinder. Thus, the sensor will measure the distance to the first occupied cell along the measurement ray, as opposed to a sensor with a wide sensing frustum (e.g., ultrasonic range sensor), which Thrun [23] handles by taking the expectation over possible obstacles in the sensing frustum. We use the notation $f_{r,n}$ to indicate the first occupied cell on the map, m^r, from the pose, x_n.

Therefore, the probability of each map, m^r, given the range measurements and corresponding poses, is

$$\begin{aligned} p(m^r|z,x) &= \frac{p(z_n|m^r, z_{1:n-1}, x)p(m^r|z_{1:n-1}, x)}{p(z_n|z_{1:n-1}, x)} \\ &= \frac{p(z_n|f_{r,n})p(m^r|z_{1:n-1}, x_{1:n-1})}{\sum_{j=1}^{R} p(z_n|f_{j,n})p(m^j|z_{1:n-1}, x_{1:n-1})}. \end{aligned} \quad (1)$$

Computing the probability of R maps is computationally expensive. In [15] we indicate that a map with $K = 20$ cells takes a few hours to compute. In this paper, we use the full Bayesian solution to which we compare the more computationally efficient algorithms. Specifically, our range sensor can trace 75 cells per measurement, so we seek to compute the full solution for 1D maps larger than this.

By assuming the rover pose (position and orientation) is fixed, we note that Equation (1) only yields K unique solutions. Specifically, $p(m^r|z, x)$ is only unique for each value of f. Therefore the R possible maps can be grouped by their respective value of f, $m^r \in f$. To compute $p(m^r|z)$, we compute the K possible solutions and then multiply by the frequency of that value of f in the denominator,

$$p(m^r|z) = \frac{p(z_n|f_r)p(m^r|z_{1:n-1})}{\sum_{j=1}^{K} 2^{(K-j)} p(z_n|f=j)p(m^j|z_{1:n-1})}.$$

Where the notation m^j refers to any $m^r \in f = j$.

This equation is written recursively to incorporate each new sensor measurement, z_n, incrementally. We assign an uninformed prior probability for each map, $p(m^r) = 1/R$. Finally, the probability of a cell being occupied is calculated by summing the map probabilities for those maps in which the cell of interest is occupied:

$$p(m_k|z) = \sum_{r=1}^{R} p(m_k|m^r)p(m^r|z), \quad (2)$$

where $p(m_k|m^r) \in \{0, 1\}$. This can be used to compare the full solution to traditional OG mapping algorithms that only calculate the probability of a cell being occupied, as opposed to the probability of a map. Note that computing the full solution is typically exponential in the number of cells in the map, K, because $R = 2^K$. But the pose-constrained formulation presented here allows us to compute the probability of only K maps. Thus, we can use this formulation to test any size map in 1D against the full Bayesian solution.

III. Traditional Occupancy Grid Mapping

Traditional OG mapping [17] makes two simplifying assumptions to reduce the computational complexity of Equation (1). The first is that the occupancy of a cell is independent of all other cells:

$$p(m) = \prod_{k=1}^{K} p(m_k).$$

This assumption allows us to compute the occupancy probability of K cells as opposed to R maps. The second is that measurements are conditionally independent, given the occupancy of a single cell of interest:

$$p(z|m_k) = \prod_{n=1}^{N} p(z_n|m_k).$$

This assumption allows us to pre-compute the update term offline. Together, these two assumptions yield a map update algorithm that is linear in the number of cells that the measurement, z_n, affects:

$$\begin{aligned} p(m_k|z,x) &= \frac{p(z_n|m_k, z_{1:n-1}, x)p(m_k|z_{1:n-1}, x)}{p(z_n|z_{1:n-1}, x)} \\ &= \frac{p(m_k|z_n, x_n)p(z_n|x_n)p(m_k|z_{1:n-1}, x_{1:n-1})}{p(m_k|x_n)p(z_n|z_{1:n-1}, x)}. \end{aligned}$$

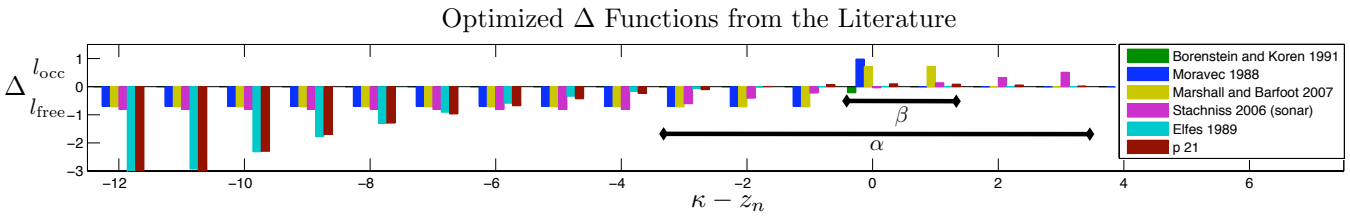


Fig. 1. Optimized update terms from five parameterizations found in the literature and our parameterization with 21 parameters (p21). The tuning parameters for each update term are indicated (l_{occ}, l_{free}, β, α) with the exception of the Elfes 1989 parameters which could not be displayed on this plot. κ is the number of cells from the robot to the cell being updated and z_n is the range measurement in units of cells.

Computation of the terms that do not depend on m_k in this expression is avoided by using the log-odds domain:

$$l(m_k|z,x) = l(m_k|z_{1:n-1}, x_{1:n-1}) + l(m_k|z_n, x_n) - l(m_k).$$

In this equation, $l(m_k)$ is a prior and $l(m_k|z_{1:n-1}, x_{1:n-1})$ is that value stored in the *evidence grid*[1] before incorporating the measurement z_n. Therefore, the key is computing the update term, $l(m_k|z_n, x_n)$, which we denote Δ. The update term is a function of the cell being updated, m_k, relative to the range measurement, z_n, and can be computed a priori. We introduce the notation, m_κ, where $\kappa = 0 \ldots F$, to refer to a cell m_k relative to the pose x_n. In other words, m_κ is the set of cells that the measurement z_n could possibly measure. Thus $m_{\kappa=0}$ is the cell that the robot occupies at pose x_n (starting pose of the measurement ray z_n) and $m_{\kappa=F}$ is the cell at the maximum range of the sensor given the position and orientation of the sensor, x_n. The update term can now be written $\Delta = l(m_\kappa|z_n)$. This update term is known as the log-odds of the inverse sensor model, $p(m_\kappa|z_n)$.

The update term governs the amount of information that is added to the map with each new measurement. However, the OG mapping literature contains several different update functions and they have not been compared or optimized to best capture the residual uncertainty in the OG map. Specifically, Figure 1 illustrates five update functions that we found in the literature and Table I specifies the tuning parameters of each.

TABLE I

THE TUNING PARAMETERS FOR VARIOUS UPDATE FUNCTIONS, Δ, FOUND IN THE LITERATURE

	Tuning Parameters	Optimized Values
Borenstein and Koren [2]	l_{occ}	-0.2197
Moravec [17]	l_{occ}, l_{free}	0.9787, -0.7021
Marshall and Barfoot [13]	$l_{\text{occ}} = -l_{\text{free}}$, β	0.7183, 2
Stachniss [21] (sonar)	l_{occ}, l_{free}, α	0.5115, -0.8135, 7
Elfes [4]	σ, $l_{\min}$	2.9627, -12.9047

The tuning parameters for each update term are shown in Figure 1. Borenstein and Koren [2] developed a reactive obstacle avoidance algorithm. To update the map quickly, they elected to use a simple update term that only adds information where the range sensor detected an obstacle, $\kappa = z_n$. In contrast, all of the other update terms add information (positive or negative) along the length of the measurement ray and some even beyond that. The update term first presented by Moravec [17] is the most commonly found in the literature [19, 21]. This update term will add l_{occ} to the evidence grid where the range sensor detected an obstacle, $\kappa = z_n$, and l_{free} to all cells between the sensor and the obstacle, $\kappa < z_n$. Arbuckle et al. [1] and Hähnel [9] used a similar update term, but enforced the constraint that $l_{\text{occ}} = -l_{\text{free}}$. Although not discussed in the paper, Marshall and Barfoot [13] also enforced this constraint but found that it resulted in the OG map overestimating the distance to the obstacle. Therefore, they allowed the update term to add l_{occ} to the evidence grid for β cells, $\kappa = z_n$ to $\kappa = z_n + \beta - 1$, which solved their issue. Stachniss [21] used two update terms in his PhD thesis. For a laser rangefinder, he used the common update term first presented by Moravec [17]. However, for a noisier sonar sensor, he linearly interpolated between l_{free} and l_{occ} with the *width* of this interpolation centered on $\kappa = z_n$. Finally, Elfes [4] presented a delta function in his PhD thesis that is based on a range sensor with Gaussian noise, $p(z_n|f) = \mathcal{N}(f, \sigma^2)$. Specifically, Elfes defines the update term as $\Delta = log \frac{\sum_{f=1}^{F} p(z_n|f)p(f|m_\kappa=1)}{\sum_{f=1}^{F} p(z_n|f)p(f|m_\kappa=0)}$. In our research, we found that thresholding the minimum value of Δ yields better results. Therefore, for this parameterization, we also include a variable, $l_{\min}$, at which we threshold the minimum value of Δ. Neither of the two parameters for [4] are shown in Figure 1 because the $l_{\min}$ value is beyond the range of the y-axis displayed and σ is used to compute Δ. In Section IV we will optimize these various tuning parameters in 1D and in Section V we will demonstrate that the optimized update terms are applicable to 2D OG mapping.

IV. Update Term Optimization

Section II explained how to compute the full Bayesian solution for OG mapping. This solution makes no assumptions, unlike the traditional OG mapping algorithms discussed in Section III. It is the goal of this paper to optimize the tuning parameters outlined in Section III for the various update terms found in the literature. We optimize these parameters in 1D where we can compute the full solution quickly by assuming the robot's pose is constrained. These optimized parameters can then be used in 2D or 3D to better capture the residual uncertainty. Finally, we optimize the update term itself for an increasing number of $\kappa - z_n$ values centered around $\kappa = z_n$.

We have previously used the Kullback-Leibler divergence to compare offline OG mapping algorithms to the full solution [15, 16]. The Kullback-Leibler divergence [11], $D_{\text{KL}}(p||q)$, quantifies the difference between two probability

[1]An evidence grid refers to an occupancy grid in the log-odds domain [14].

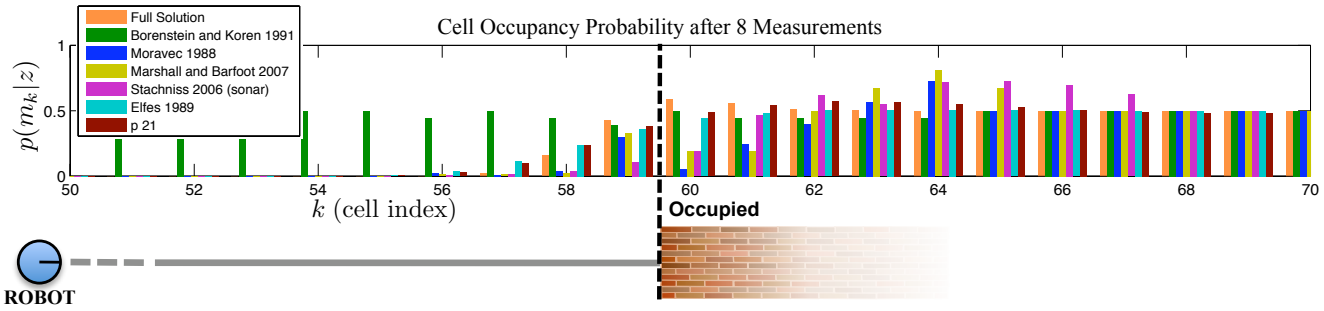


Fig. 2. The occupancy grids computed by the various algorithms in the literature and our parametrization with 21 parameters for one of the 1D datasets. The full solution is also shown for reference. Note that our algorithm most closely resembles the full solution near the obstacle, $f = 60$.

distributions, p and q, where p is the benchmark probability, and q is the estimated probability. For the purpose of comparing two occupancy grids, we compute $D_{\mathrm{KL}}(p_k||q_k)$ for each cell, m_k, in the map between the full solution and estimated map. Once

$$D_{\mathrm{KL}}(p_k||q_k) = \sum_{m_k=0}^{1} p(m_k) \log \frac{p(m_k)}{q(m_k)}$$

is computed for each cell, m_k, the sum over all cells is the D_{KL} between the two maps:

$$D_{\mathrm{KL}}(p||q) = \sum_{k=1}^{K} D_{\mathrm{KL}}(p_k||q_k).$$

However, to optimize the tuning parameters, we need only optimize the cross entropy between the two maps,

$$H(p_k, q_k) = -\sum_{m_k=0}^{1} p(m_k) \log q(m_k),$$

$$H(p, q) = \sum_{k=1}^{K} H(p_k, q_k),$$

because the cross entropy and Kullback-Leibler divergence are equal to within an additive constant if the benchmark probability, p, is constant. This is known as the *principal of minimum discrimination information* [10].

Each of the tuning parameters for the five update terms detailed in Section III were optimized using the BFGS [3, 6, 8, 20] Quasi-Newton method with a mixed quadratic and cubic line search procedure. Specifically, we used the same 1D dataset with 1000 OG maps as we did in [15] and [16]. However, we divided the 1000 datasets into 10 equal sets and performed 10-fold cross validation. In other words, we optimized on 900 datasets, then tested on 100 datasets and repeated this procedure 10 times. In this optimization, the range sensor had a maximum range of $F = 75$ cells, but the obstacle was $f = 60$ cells from the (static) robot and the range sensor had Gaussian noise with a standard deviation of $\sigma = 3$ cells, $p(z|f) = \mathcal{N}(60, 3^2)$. Each dataset produced an OG map using traditional two assumption OG mapping and the tuning parameters being optimized for the update term. This occupancy grid was compared to the full solution using cross entropy to quantify the divergence of the two maps. The best update terms from this optimization are shown in Figure 1 and the third column of Table I. Figure 2 illustrates a portion of the OG created by each of the optimized algorithms for one of the 1000 datasets. Figure 2 also shows the full Bayesian solution for reference.

Additionally, Figure 1 and Figure 2 show the optimized update term for "p21" and an example OG created by "p21" respectively. The "p21" algorithm refers to an optimization that we conducted by optimizing an increasing number of $\Delta(\kappa - z_n)$ values centered around $\kappa = z_n$. Visually, this can be seen as optimizing each bar in Figure 1 where each bar (value of Δ) is a parameter in the optimization. Since we optimized an increasing number of parameters, we assumed that $\Delta = 0$ for values beyond (away from the robot) the parameters being optimized and we assumed that Δ is equal to the first value being optimized (nearest to the robot) for the values of delta that were closer to the robot than those being optimized. For one parameter being optimized, p1, our method outperformed Borenstein and Koren [2] because we were able to add information to all cells between the rover and the obstacle. However, our method is equivalent to Moravec [17] for two parameters, p2. Thus, as we increase the number of parameters, this method should be able to capture and outperform all of the other algorithms in the literature. Figure 3 shows the optimized parameters for the increasing number of parameters. It is interesting to note that as the number of parameters increases the update term becomes more negative for values of $\kappa < z_n$. Also, near $\kappa = z_n$ the update term adds less information per cell but distributed over more cells as the number of parameters increases.

Figure 4 shows how the cross entropy decreased as we increased the number of parameters being optimized. Figure 4 also shows the best cross entropy values for the other algorithms in the literature for reference; the parametrization from [2] was excluded from Figure 4 because its cross entropy values were significantly higher than the rest. It is interesting to note that cross entropy is more dramatically reduced as we add parameters near the robot as opposed to beyond the obstacle. Intuitively this makes sense as the range sensor should not be changing the occupancy probability of cells beyond the obstacle. Furthermore, Figure 4 shows that the parametrization proposed by Elfes [4] did extremely well and our method had to optimize 19 or more parameters to do better. Figure 1 gives more insight into this result as we see where Elfes 1989 and p21 differ. Figure 5 further

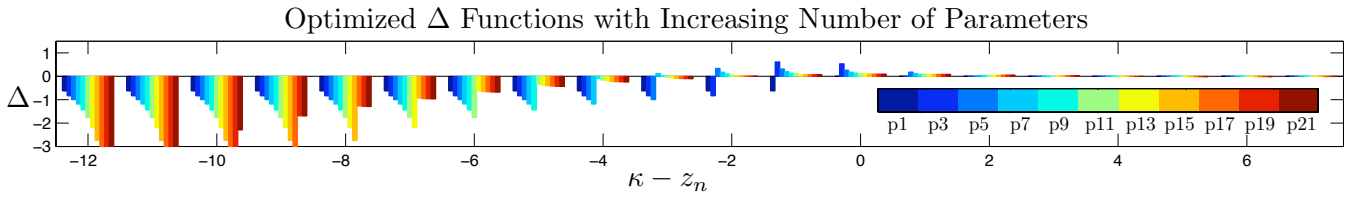


Fig. 3. The optimized update term for an increasing number of parameters centered around $\kappa = z_n$. The optimization was completed for parameters 1 to 21, but only every other parametrization is illustrated here.

highlights this result by illustrating the average D_{KL} on a cell-by-cell basis for the cells near the obstacle. These figures show that our parameterization better models the full solution near the obstacle at the expense of doing slightly worse beyond the obstacle. We clearly see that the algorithms (with the exception of [2]) diverge most from the full solution near the obstacle and that our parametrization of the update function diverges the least. Particularly, the parametrization proposed by Moravec [17] diverges the most near the obstacle boundary and this is the most common OG mapping algorithm found in the literature.

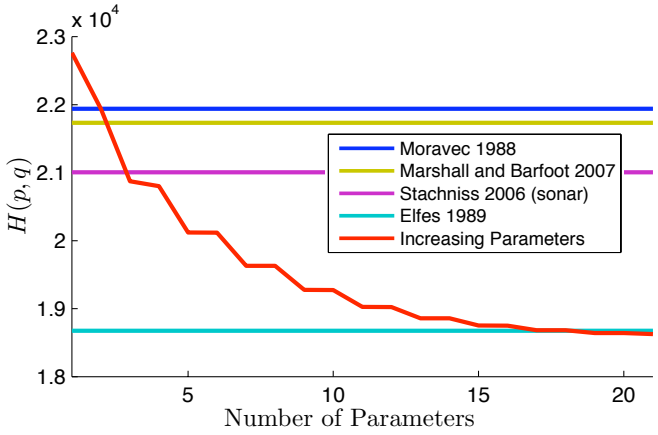


Fig. 4. Best cross entropy values for optimized two-assumption OG mapping algorithms as compared to the full Bayesian solution.

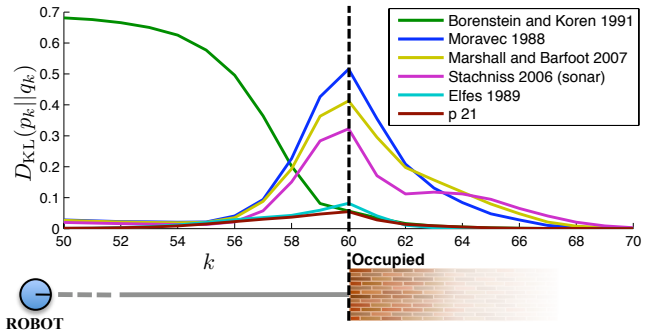


Fig. 5. The average D_{KL} for each of the two-assumption OG mapping algorithms in the literature and our solution with 21 parameters, compared to the full Bayesian solution. The obstacle is at $k = 60$, so we see that the largest divergence is centered around the obstacle and that our parametrization has the lowest average divergence.

V. 2D Results

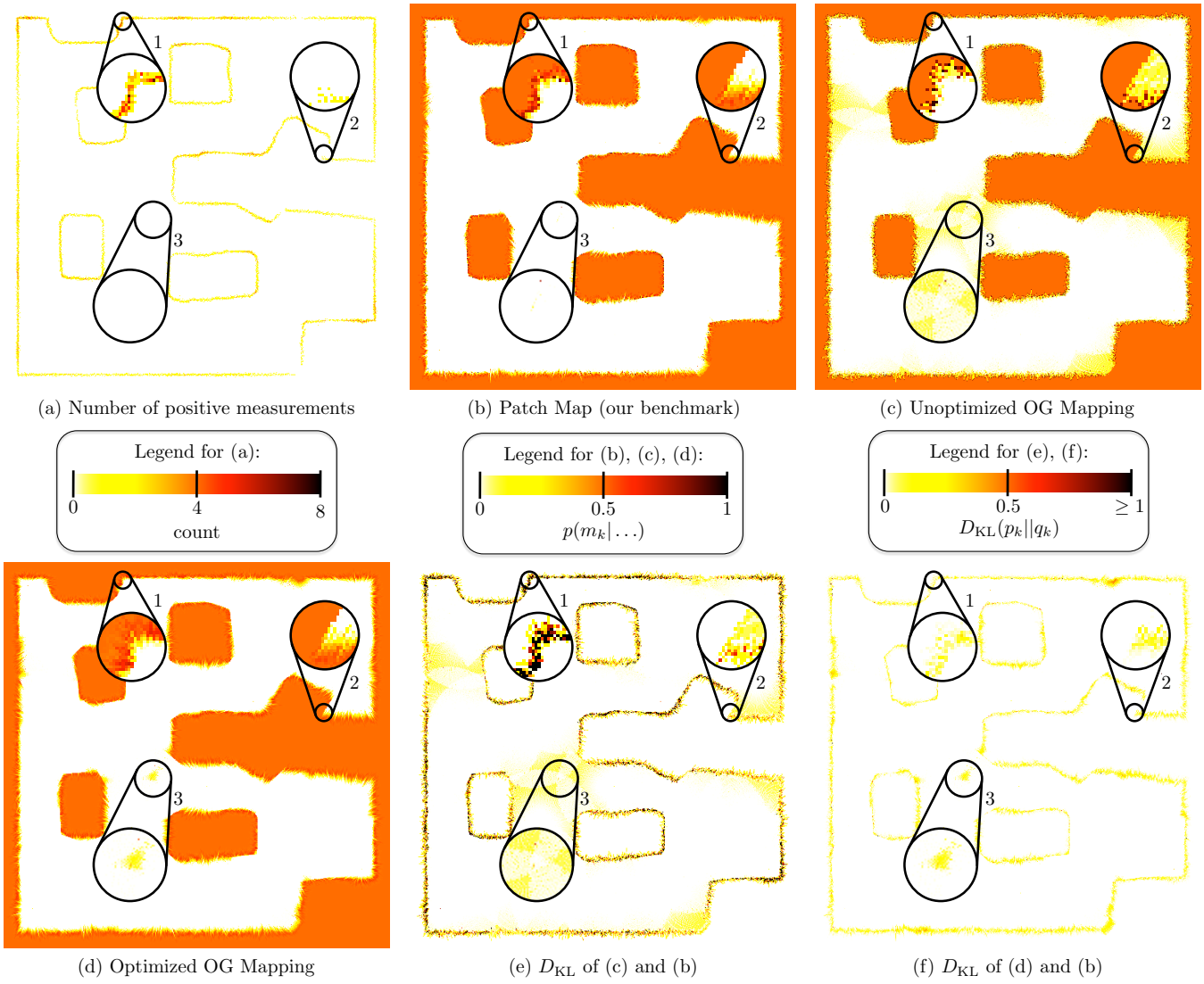
To show that the optimized tuning parameters from Section IV perform well in realistic scenarios, we have tested the optimized parameters from the literature and the 21-parameter update term on 100 2D simulations. The 2D data was collected on 10 different maps that range from cluttered office-type environments to sparsely occupied maps. We ran 10 mapping experiments on each of the 10 maps using an exploration algorithm that ensured all areas of the environment were mapped. The robot only mapped the environment when stopped, as is common with exploration algorithms that seek to produce highly accurate maps [24]. Each map was 500×500 cells and the laser rangefinder could measure a maximum of $F = 75$ cells with Gaussian noise on each measurement, $p(z_n|f) = \mathcal{N}(f, 3^2)$.

The full Bayesian solution cannot be tractably computed for these large 2D maps. Therefore, we computed the patch map [15] for each of the 2D maps. The patch map is an offline algorithm that has been shown to accurately capture the residual uncertainty in an occupancy grid and therefore approximate the full Bayesian solution. Essentially, the patch map computes the full solution for a small patch of cells and uses that result to report the occupancy probability of a cell. We used a 3×3 patch size ($W = 9$) and the maximum a posteriori (MAP) estimate for the patch map algorithm.

Once the update term parameters have been optimized, the new update term can be used online. Therefore, the computation time is the same as existing methods. For comparison, we used the update term presented by Hähnel [9]. This update term can be seen as the parameterization presented by Moravec [17] or Marshall and Barfoot [13] as Hähnel used $l_{\mathrm{occ}} = -l_{\mathrm{free}} = 1.3863$ as parameters[2]. Table II shows the average D_{KL} for each set of parameters compared the to the patch map. As expected, we see that the optimized parameters of [17] and [13] outperform the unoptimized parameters presented in [9]. Using the unoptimized parameters, the average D_{KL} between the maps produced and the patch map was 1.47×10^4. Furthermore, we see that our parameterization with 21 parameters has the lowest average D_{KL} compared to the patch map with a value of 3.97×10^3. Thus, the new update term allowed us to better approximate the offline patch map method.

Figure 6 illustrates the improved performance of the OG mapping algorithm by using the optimized delta function as opposed to the parameters found in the literature. The figure compares the performance on one of the 100 2D datasets. Image (a) shows the frequency with which each cell returned a range reading. The three black circles on each image are magnifications. Note that the first magnification shows a heavily-mapped area and the second shows an under-mapped area. Image (b) shows the reference patch map. Image (c)

[2]Hähnel [9] reports these values in the probability domain as $p_{\mathrm{free}} = 0.2$ and $p_{\mathrm{occ}} = 0.8$.



(a) Number of positive measurements (b) Patch Map (our benchmark) (c) Unoptimized OG Mapping

(d) Optimized OG Mapping (e) D_{KL} of (c) and (b) (f) D_{KL} of (d) and (b)

Fig. 6. Results from one of the 100 2D datasets. Image (a) shows the number of times the range sensor reflected off a cell, m_k, in the dataset. Image (b) illustrates the patch map [15] using the MAP estimate. Image (c) is the OG created using the update term in [9]. Image (d) is the OG created using the optimized 21-parameter update term. Image (e) displays the D_{KL} between (c) and (b) and image (f) displays the D_{KL} between (d) and (b). In both images (e) and (f) the values have been thresholded at $D_{\mathrm{KL}}(p_k||q_k) = 1$ for the purpose of this illustration. The circles in the six images are magnifications. The first magnification shows that the traditional OG mapping algorithm is overconfident in well-mapped areas. The second magnification shows that the optimized parameters out-perform the traditional method, even on walls with few measurements. The third magnification shows that the optimized parameters also estimate the unoccupied cells more accurately. Each cell in the third magnification received few (between one and three) measurements.

TABLE II

D_{KL} BETWEEN ONLINE OG MAPPING AND PATCH MAP IN 2D

	mean(D_{KL})	min(D_{KL})	max(D_{KL})
Hähnel [9]	1.47×10^4	8.91×10^3	2.37×10^4
Borenstein and Koren [2]	1.25×10^5	1.07×10^5	1.37×10^5
Moravec [17]	9.57×10^3	6.29×10^3	1.39×10^4
Marshall and Barfoot [13]	1.12×10^4	7.22×10^3	1.67×10^4
Stachniss [21] (sonar)	7.60×10^3	4.78×10^3	1.19×10^4
Elfes [4]	4.18×10^3	2.58×10^3	6.53×10^3
p 21	3.97×10^3	2.62×10^3	5.79×10^3

shows the OG using parameters from the literature and image (d) shows the OG using the optimized parameters. Images (e) and (f) show the D_{KL} on a cell-by-cell basis between (c) and (d), respectively, compared to the patch map. The many dark cells in image (e) indicate that the OG created using parameters from the literature strongly diverges from the patch map, especially in heavily-mapped areas. Note that the largest divergence is concentrated in areas where the range sensor returned a reading, as seen in image (a). Furthermore, the third magnification highlights an unoccupied area. We see that the optimized parameters better capture the residual uncertainty in this area as well. Each of the cells in the third magnification had only one to three measurements. Figure 1 shows that our parametrization of the update term reduces the occupancy of cells in the unoccupied areas more than others. Therefore, it is no surprise that our update term outperforms the others in unoccupied areas.

VI. Conclusion and Future Work

This paper highlights that existing OG mapping techniques do not always capture the residual uncertainty in the map

as well as they could. By comparing the full Bayesian solution in 1D and the patch map in 2D, we see the benefit of using a better parametrization of the update term and optimizing those parameters to capture the uncertainty. We compared several update terms that have been presented in the literature and also presented our own parametrization. Of those found in the literature, the parameterization first presented by Elfes [4] stood out as the best. However, the majority of OG mapping algorithms in the literature are using the parameterization first presented by Moravec [17]. Our own parameterization was able to outperform those in the literature. Figure 4 may lead the reader to believe that our performance increase was minor, but Figures 2 and 5 illustrate that our update term yields better accuracy at the threshold from unoccupied to occupied at the cost of performing slightly worse beyond the first-occupied-cell. This improvement comes at no additional computational cost during online operations.

The optimized update terms presented in this paper can be used online for any two-assumption OG mapping algorithm. This is unlike our previous work to capture the residual uncertainty in an occupancy grid [15, 16] that was for offline use. It should be noted, however, that the offline algorithms still perform better than this online algorithm as they account for correlations between cells in the map. Future work on this topic will further compare these online and offline methods. In addition, better representing the residual uncertainty in the map should allow us to improve the common uses of OG maps, such as exploration and localization.

ACKNOWLEDGMENT

The authors thank Dr. Joseph Bakambu from MDA Space Missions for his valuable contribution to this work.

REFERENCES

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

[2] J. Borenstein and Y. Koren, "The vector field histogram - fast obstacle avoidance for mobile robots," *IEEE Journal of Robotics and Automation*, vol. 7, no. 3, pp. 278–288, June 1991.

[3] C. Broyden, "The convergence of a class of double-rank minimization algorithms," *Journal Inst. Math. Applic.*, vol. 6, pp. 76–90, 1970.

[4] A. Elfes, "Occupancy grids: A probabilistic framework for robot perception and navigation," Ph.D. dissertation, Carnegie Mellon University, 1989.

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

[6] R. Fletcher, "A new approach to variable metric algorithms," *Computer Journal*, vol. 13, pp. 317–322, 1970.

[7] S. Geman and D. Geman, "Stochastic relaxation, gibbs distribution, and the bayesian restoration of images," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721–741, 1984.

[8] D. Goldfarb, "A family of variable metric updates derived by variational means," *Mathematics of Computing*, vol. 24, pp. 23–26, 1970.

[9] D. Hähnel, "Mapping with mobile robots," Ph.D. dissertation, University of Freiburg, Department of Computer Science, December 2004.

[10] S. Kullback, *Information Theory and Statistics*. New York, NY: John Wiley, 1959.

[11] S. Kullback and R. A. Leibler, "On information and sufficiency," *Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, March 1951.

[12] A. Makarenko, S. Williams, F. Bourgault, and H. Durrant-Whyte, "An experiment in integrated exploration," *IEEE/RSJ Int. Conf. on Intelligent Robots and System (IROS)*, vol. 1, pp. 534–539, 2002.

[13] J. Marshall and T. Barfoot, "Design and field testing of an autonomous underground tramming system," in *Int. Conf. on Field and Service Robotics (FSR)*, July 2007.

[14] M. C. Martin and H. P. Moravec, "Robot evidence grids," Carnegie Mellon University Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-96-06, March 1996.

[15] R. S. Merali and T. D. Barfoot, "Patch map: A benchmark for occupancy grid algorithm evaluation," *IEEE/RSJ Int. Conf. on Intelligent Robots and System (IROS)*, pp. 3481–3488, October 2012.

[16] ——, "Occupancy grid mapping with Markov chain monte carlo Gibbs sampling," *Int. Conf. on Robotics and Automation (ICRA)*, pp. 3168–3174, May 2013.

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

[18] H. Moravec and A. E. Elfes, "High resolution maps from wide angle sonar," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, March 1985, pp. 116–121.

[19] D. Murray and J. J. Little, "Using real-time stereo vision for mobile robot navigation," *Autonomous Robots*, vol. 8, no. 2, pp. 161–171, 2000.

[20] D. Shanno, "Conditioning of quasi-newton methods for function minimization," *Mathematics of Computing*, vol. 24, pp. 647–656, 1970.

[21] C. Stachniss, "Exploration and mapping with mobile robots," Ph.D. dissertation, University of Freiburg, Department of Computer Science, April 2006.

[22] C. Stachniss and W. Burgard, "Exploring unknown environments with mobile robots using coverage maps," *Int. Conf. on Artificial Intelligence (IJCAI)*, vol. 1, pp. 1127–1132, 2003.

[23] S. Thrun, "Learning occupancy grids with forward sensor models," *Auton. Robots*, vol. 15, pp. 111–127, 2003.

[24] C. Tong, T. D. Barfoot, and E. Dupuis, "Three-dimensional SLAM for mapping planetary worksite environments," *Journal of Field Robotics,* special issue on "Space Robotics", vol. 29, no. 3, pp. 381–412, 2012.