2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

MCMC Occupancy Grid Mapping with a Data-Driven Patch Prior

Rehman S. Merali and Timothy D. Barfoot

***Abstract*— Occupancy grids have been widely used for mapping with mobile robots for several decades. Occupancy grids discretize the analog environment and seek to determine the occupancy probability of each cell. More recent occupancy grid mapping algorithms have shown the advantage of capturing cell correlations in the measurement model and the posterior. By estimating the probability of a given map as opposed to a cell, these algorithms have been able to better capture the occupancy probability of cells in the map. The advantage of incorporating data-driven prior probabilities in occupancy grid mapping is explored. A form of Markov Chain Monte Carlo (MCMC) known as Gibbs sampling allows us to sample maps from the full posterior. Previous research has sampled the occupancy probability of each cell, but this paper extends that work to sample a larger *patch* of cells and highlights the benefit of obtaining the prior for each *patch* from real maps.**

I. Introduction

Mobile robots often make strong assumptions about the environment to generate real-time maps. This paper presents an *anytime* method to improve the map estimate when additional computation time is available. Figure 1 shows that this method is better able to estimate the ground truth map than traditional methods using the same set of measurements.

Occupancy grid mapping (OGM) has been a popular online mapping technique for over 30 years [7, 22, 23]. It represents the occupancy of each cell with a binary random variable and computes the occupancy probability of each cell. Traditionally, OGM assumes cells are independent. This paper will show that neighbouring cells are not necessarily independent and that correlating neighbouring cells can result in OG maps that are better able to estimate the true map.

This paper will review related work in Section II and Section III will present a novel OGM algorithm that correlates proximate cells in the prior. Section IV will illustrate the benefit of the novel algorithm by comparing the resulting OG map on simulated and hardware data to traditional OGM methods. This section will highlight the computational benefit of this method and its increased accuracy, especially when measurement data is sparse. Finally, Section V will conclude the paper and suggest possible future extensions.

II. Related Work

A popular approach to representing occupancy in the map without a grid is a Gaussian Processes (GP) [29], which is a stochastic process that enables high-dimensional regression by using measurements to learn covariance and mean functions. GPs enable Continuous Occupancy Mapping (COM) that estimates a map as a continuous smooth surfaces [25, 26]. Gaussian process occupancy mapping (GPOM) [26] is able to estimate the occupancy of areas in the map without a grid structure, but a major drawback is its cubic time complexity. GPOMs are able to estimate unmapped areas, but require batch offline optimization of hyper-parameters and require selection of an appropriate prior, although recent research has reduced the computational cost [14, 17]. Hilbert maps [10, 28, 31, 35] are continuous occupancy maps that scale more efficiently than GPOMs.

Both authors are with the University of Toronto Institute for Aerospace Studies (UTIAS), 4925 Dufferin St., Toronto, Canada: rehman.merali@utoronto.ca, tim.barfoot@utoronto.ca

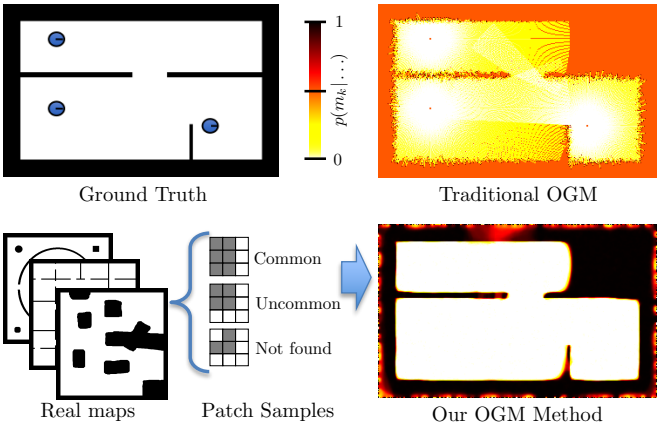


Fig. 1: OGM seeks to estimate a map from range measurements. In this example, measurements are taken from three robot locations (blue circles). Our method correlates cells in the prior using an MCMC Gibbs Sampling algorithm. Patch samples are drawn from real maps to estimate unmapped cells between mapped cells and produces straighter walls using the same noisy range measurements.

Occupancy grid mapping was originally developed as a binary Markov Random Field (MRF), where each cell is considered independent. In the conclusion of his PhD thesis, Elfes [7] recognized that "the application of higher-order Markov Random Field estimation methods to Occupancy Grids needs to be investigated." Recently, Muffert [24] and Shankar and Michael [32] have explored a higher-order MRF in OGM to model dependencies between neighbouring grid cells using cameras. Shankar and Michael used loopy belief propagation to correlate cells along measurement rays. In addition, recent research has used a kernel inference model to estimate unobserved areas of an occupancy grid using nearby range measurement [6, 16]. In this paper, we seek to correlate cells in both the prior and along measurement rays, and to represent these correlations in the posterior.

Traditional OGM has been optimized using variational inference to develop an online solution that better captures

the residual uncertainty in an OG map [20]. Furthermore, a batch offline solution that computes the full Bayesian solution for a *patch* of neighbouring cells was developed in [18]. An anytime Markov Chain Monte Carlo (MCMC) algorithm has been shown to draw samples from the full Bayesian Solution [19]. This MCMC algorithm was then extended to a factor graph approach [4, 5] and also used for viewpoint-based exploration [12]. This paper further extends this MCMC algorithm by introducing cell correlations in the prior. The method presented is not limited to orthogonal structures, as is other research on mapping priors [2, 30, 33].

III. MCMC with Data-Driven Prior

Occupancy grid mapping discretizes an analog environment, m, into a regular grid of K cells. 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 occupancy grid 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. 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.

Several proposed methods account for range measurement noise [1, 4, 15], but generally occupancy grid mapping algorithms use Bayesian methods to compute the probability that each cell is occupied, $p(m_k)$, given a set of 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 previously presented the full solution to OGM to compute the probability of each map given the set of range measurements and corresponding robot poses, $p(m^r|z, x)$ [18, 19]. However, this paper questions the assumed uniform prior map probability, $p(m^r) = 1/R$.

The full Bayesian solution to OGM is computationally intractable for realistic maps. Therefore, traditional OGM makes two simplifying assumptions to reduce the computational complexity [22]: a cell independence assumption and a conditional measurement independence assumption,

$$p(m) = \prod_{k=1}^{K} p(m_k), \quad \text{and} \quad p(z|m_k) = \prod_{n=1}^{N} p(z_n|m_k).$$

Together, these two assumptions yield a map update algorithm that is linear in the number of cells, which the measurement, z_n, affects:

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

The details of applying this equation are found in [20], but for this paper we use the inverse sensor model, $p(m_k|z_n, x_n)$, used by Hähnel [11] as $p_{\text{free}} = 0.2$ and $p_{\text{occ}} = 0.8$, and refer to this as *traditional OGM*.

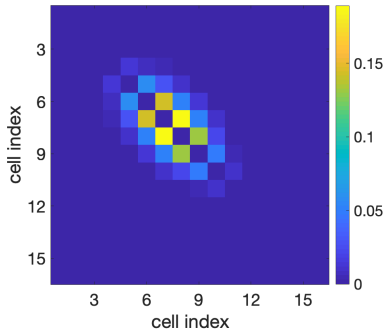


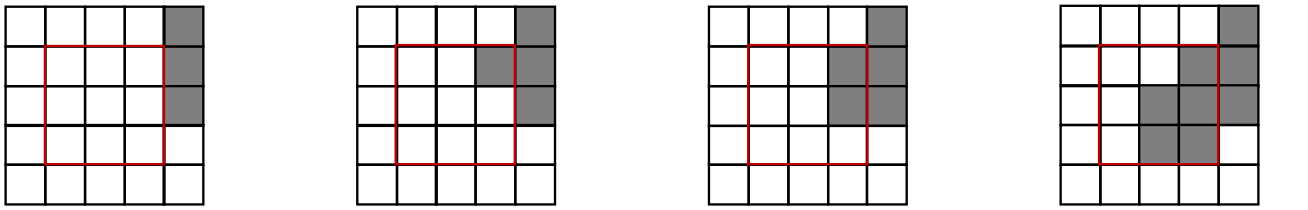
Fig. 2: Mutual information between cells using the full Bayesian solution on a 1D toy example. The robot is at cell $k = 1$ and the first-occupied-cell is at $f = 8$. Traditional OGM assumes that this information is zero.

Figure 2 shows the mutual information between every pair of cells using the full Bayesian solution for a 1D toy example where the robot takes six range measurements with Gaussian random noise about the true first-occupied-cell at $f = 8$. The figure highlights that mutual information is greatest between proximate cells and approaches zero for cells that are farther apart. The cell independence assumption in traditional OGM discards this mutual information by assuming it is zero. The MCMC Gibbs Sampling algorithm developed in [19] is able to capture the mutual information in OGM by drawing samples from the full posterior. This anytime algorithm is able to approximate the otherwise intractable full solution with increasing fidelity as more samples are drawn. This paper introduces cell correlations in the prior by generalizing that algorithm to sample a patch of cells, a_k, centered on the cell k as opposed to a single cell, m_k,

$$\begin{aligned} &p(a_{k,w}|z, x, m_{\neg a_k}) \\ &\quad = \frac{p(a_{k,w}|x, m_{\neg a_k})p(z|x, m_{\neg a_k}, a_{k,w})}{p(z|x, m_{\neg a_k})} \\ &\quad = \eta \underbrace{p(a_{k,w}|m_{\neg a_k})}_{\text{patch prior}} \prod_{n=1}^{N} \underbrace{p(z_n|x_n, m_{\neg a_k}, a_{k,w})}_{\text{sensor model}}, \end{aligned} \quad (1)$$

where η is a normalizing constant and the term $m_{\neg a_k}$ represents the occupancy of all cells in the map, m, except those in the patch a_k. Equation (1) makes the static-world assumption that measurements are independent of one another given the occupancy of all cells. This assumption is quite different than the one made in traditional OGM in which measurements are considered independent given the occupancy of one cell.

Algorithm 1 shows the MCMC Gibbs sampling algorithm, which generalizes the algorithm presented in [19] to sample patches of L cells using Equation (1). Figure 3 shows examples of 3×3 (in red) and 5×5 patches. Section III-A will show that some patches may have a prior probability of zero; therefore, Algorithm 1 includes variable $p_{\text{randPatch}}$ to ensure the algorithm is ergodic. In this paper, we set $p_{\text{randPatch}} = 0.001$ to balance the time to reach a stationary distribution while still exploring the solution space. After reaching a stationary distribution, the algorithm returns map samples from the full posterior. These map samples can be used to compute various statistics such as the occupancy



(a) $p(a_{k,w}) = 74.30\%$, $p(a_{k,w}|\text{border}_{a_k}) = 97.41\%$ (b) $p(a_{k,w}) = 0.0473\%$, $p(a_{k,w}|\text{border}_{a_k}) = 1.87\%$ (c) $p(a_{k,w}) = 0.0134\%$, $p(a_{k,w}|\text{border}_{a_k}) = 0.58\%$ (d) $p(a_{k,w}) = 0.0103\%$, $p(a_{k,w}|\text{border}_{a_k}) = 0.13\%$

Fig. 3: An example of four 3×3 patches ($L=9$) and their likelihood given the same 16 border cells. Conditioning the patch prior on the border results in a significant computational savings because so few (often one) patch priors are nonzero. The four 3×3 patches depicted (in red) are the only four sampled from the data with the given border.

Algorithm 1 MCMC Gibbs sampling with a data-driven patch prior

Given z, x, L, $p_{\text{randPatch}}$, $m^{(0)}$, maxSamples, burnIn
Define m^r as $m^{(0)}$
for $i = 1$ to maxSamples **do**
  for $t = 1$ to K/L **do**
    Select random patch a_k from m^r
    if rand(0,1) $< p_{\text{randPatch}}$ **then**
      Set $a_k^{(i)}$ to a random patch configuration
    else
      Sample $a_k^{(i)}$ from $p(a_{k,w}|z, x, m^r_{\neg a_k})$
    end if
    Define a_k^r, in m^r, as $a_k^{(i)}$
  end for
  Set $m^{(i)} = m^r$
end for
return set of samples, $m^{\text{burnIn:maxSamples}}$, from the full posterior

probability of a cell or group of cells by averaging their occupancy over many samples. A stationary distribution is quickly reached if the algorithm begins with a thresholded OG map developed using traditional methods.

Equation (1) is used to draw patch samples, $a_k^{(i)}$, in Algorithm 1 and this equation requires a forward sensor model, $p(z_n|f)$, and a patch prior, $p(a_{k,w}|m_{\neg a_k})$. This paper assumes a narrow-beam range sensor with Gaussian noise about the true measurement, $p(z_n|f) = \mathcal{N}(f, 3^2)$. The patch prior will be explored in the following two subsections.

A. Data-Driven Prior

In this subsection, we assume that the patch prior in Equation (1) is independent of other cells in the map, $p(a_{k,w}|m_{\neg a_k}) \approx p(a_{k,w})$. To compute this term, 5×5 patches were randomly drawn from realistic maps with $K = 250\,000$ (500×500) cells. These ten maps [21] were designed to provide a variety of building-type and unstructured environments with walls of varying shape, thickness and distance between them. This variety also provided a diverse set of patch priors. Each patch sample was flipped and rotated to capture all eight permutations to remove orientation bias. The 10^{10} patch samples drawn were then subsampled for other cell configurations, such as 3×3 patches or single cells. These samples showed that the probability of a cell being occupied is $p(m_k) = 0.234$, which is also the average number of occupied cells in the ten maps.

Many patch configuration were not observed in the samples. For example, of the $2^9 = 512$ possible patch configurations with a patch size of $L = 9$, only 124 patch configurations were ever sampled and 99.9% of those samples were of the 50 most likely patch configurations. Furthermore, 74.3% of 3×3 patches sampled were fully unoccupied and 21.1% were fully occupied and therefore together accounted for $>95\%$ of samples. Similarly, for a patch size of $L=25$, only 2329 possible patch configurations (of 2^{25}) were ever sampled, which is 0.0069% of the patch configurations. Additionally, 99.9% of samples drawn were of the most likely 436 patch configurations. 71.98% of the sampled 5×5 patches are fully unoccupied and 18.92% are fully occupied; therefore these two patch configurations together account for $> 90\%$ of samples. Figure 3 shows the probability of four 3×3 patches (in red) and highlights that occupied cells tend to be clustered together in realistic patches.

The results in this paper, including Table I, were computed using unoptimized Matlab code on a computer running Ubuntu with 32GB of RAM and a 2.70GHz Intel i7-6820HQ CPU. This table shows the average computation time for the 2D toy example seen in Figure 1. These results show that a data-driven prior is slower than a uniform prior when $L=1$ because the computation is not carried out in the log-odds domain. However, as the patch size increases to $L = 9$ the data-driven prior method is able to draw map samples faster than the uniform prior method. As the patch increases to $L = 25$, this result is more pronounced as the method is computationally intractable for a uniform prior, but tractable with a data-driven prior.

B. Data-Driven Prior with Border

In this section, the patch prior is conditioned on the cells that border the patch, $p(a_{k,w}|m_{\neg a_k}) \approx p(a_{k,w}|\text{border}_{a_k})$. The 5×5 patch samples from Section III-A are used to compute the probability of 3×3 patch configurations, given their 16 bordering cells, and of individual cells given their 8 bordering cells. Figure 3 shows an example 16-cell border and highlights the probability of the only four 3×3 patch configurations given that border. The increased sparsity of relevant patches leads to greater computational efficiency and accuracy. For instance, note that a fully occupied patch has zero probability and that three of the four patch configurations are asymmetric as occupied cells are clustered together.

In analyzing the 10^{10} patches, only 865 of the 2^{16} borders with 16-cells are observed. The majority of borders have only one unique 3×3 patch configuration and the maximum is 13; much less than the 124 patch configurations observed in the data. Therefore, conditioning the patch prior on the border in Equation (1) yields a significant computational efficiency as very few patch configurations have a nonzero likelihood given the border. In most cases, the patch prior is zero for all but one patch configuration and therefore the measurements, $p(z_n|x_n, m_{\neg a_k}, a_{k,w})$, have no effect on those cases and the MCMC algorithm will sample the one patch configuration with a nonzero prior probability in Algorithm 1.

Figure 4 shows that the average probability of the most likely 3×3 patch given its border is 0.865 and that the four most likely patches account for 0.997 of the probability mass. The figure also illustrates (in red) the maximum and minimum prior probability of accounting for additional 3×3 patch configurations. Therefore, conditioning the 3×3 patch prior on its border cells, reduces the computation to at most 13 patch configurations, but most of the benefit can be achieved by computing even fewer patch configurations. The 10^{10} patches are also used to determine the probability of a cell, $L = 1$, given its 8 bordering cells. This analysis shows that only 84 of the possible $2^8 = 256$ borders are observed in the data. Of these, only 40 borders contain both an occupied and unoccupied cell and the remaining 44 are perfectly confident in the occupancy of the cell being sampled by the MCMC algorithm.

Table I shows that the border-based data-driven prior is slower for $L = 1$ because of the added computational cost to look up a unique prior value based on the border. However, the computational benefit of this method is seen with larger patch sizes where the border limits the possible patch configurations. While a patch size of $L = 25$ would yield significant sparsity, it was not computed because 7×7 patches were not drawn to compute the border-based prior values. However, for a patch size of $L = 9$, this algorithm is able to compute a map sample (i.e., K/L patches) with a border-based data-driven prior faster than a uniform prior with $L = 1$ as presented in [19]. Figure 1 shows the OG map generated by averaging the cell marginals over 5000 samples. This map more accurately resembles the ground truth map than traditional OGM. Specifically, this method is able to estimate unmapped cells between measurement rays and produce straight walls given the same measurement data.

TABLE I: Average time to compute a map sample (K/L patches) for various patch sizes, L, and priors. The 2D map has $K = 64\,000$ cells. A data-driven prior (especially when conditioned on the border) is more efficient because few configurations have a nonzero prior probability.

	Uniform Prior [s]	Data-Driven Prior [s]	Data-Driven Prior with Border [s]
$L=1$	**4.4**	5.1	6.6
$L=9$	38.3	23.9	**2.7**
$L=25$	intractable	**124.9**	

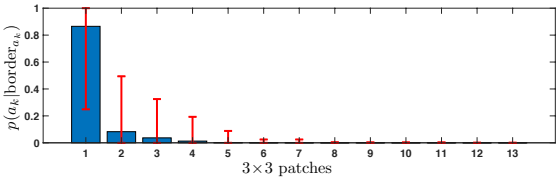


Fig. 4: Average probability of a 3×3 patch, given its 16-cell border. The error bars indicate the range of possible values, highlighting how few patches need to be considered.

IV. Experimental Results

To highlight the benefit of using a data-driven prior, the MCMC algorithm was compared to traditional OGM on simulation and hardware data. The simulated data includes 100 2D datasets (10 runs on each of 10 realistic maps) with a map size of 500×500 cells, range measurements with $F = 75$ cells and Gaussian measurement noise, $p(z_n|f) = \mathcal{N}(f, 3^2)$. A simple exploration algorithm is used to ensure coverage for each dataset. The robot only maps the environment when stopped, as is common with exploration algorithms that seek to produce highly accurate maps [34].

To evaluate the experiments in this paper, the 2D OG maps generated are thresholded and compared to the binary ground truth map in simulation and a binary benchmark map in hardware. Georgiou et al. [9] propose the use of *precision* and *recall* [3] to compare the maps on a cell-by-cell basis. In this paper, we also use *precision* and *recall* to evaluate OG maps, but weight the two equally with an F_1 metric, as opposed to an F_2 metric that favours recall over precision. To generate these metrics, each cell in the OG map is marked as a true positive (TP) if it matches the benchmark map, a false positive (FP) if it estimates an unoccupied cell as occupied, and as a false negative (FN) if it estimates an occupied cell as unoccupied. The two metrics are therefore defined as [3],

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{and} \quad \text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}.$$

The F_1 score is a metric often used to balance these two factors as a weighted sum,

$$F_1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \tag{2}$$

For the four algorithms tested in simulation, Figure 5 illustrates the precision-recall curves over all 100 simulated datasets, and Table II summarizes the highest precision, recall, and F_1 score achieved for each method. Furthermore, Figure 6 plots the F_1 score over all 100 datasets for various threshold values. This analysis was conducted by estimating the maps for all 100 simulated datasets and varying the threshold value for each algorithm's estimate of the marginal cell occupancy for all $500 \times 500 \times 100$ cells. Not only does the border-based algorithm have the highest precision and F_1 score, but Figure 6 highlights that it is robust to the threshold as there is little uncertainty in the marginal occupancy probability of each cell. Note that traditional OGM has a large discontinuity at the cell prior because

TABLE II: Precision, Recall and F_1 of various algorithms on 100 simulated datasets.

	Traditional OGM	MCMC, $L=1$	MCMC, $L=9$	MCMC +border, $L=9$
Max Precision	0.9717	0.9571	0.9913	**0.9976**
Max F_1 Score	0.9328	0.9460	0.9458	**0.9820**
Threshold for max F_1	0.20	0.10	0.10	0.07
Precision @ max F_1	0.9310	0.9026	0.9032	**0.9694**
Recall @ max F_1	0.9346	0.9939	0.9925	**0.9950**

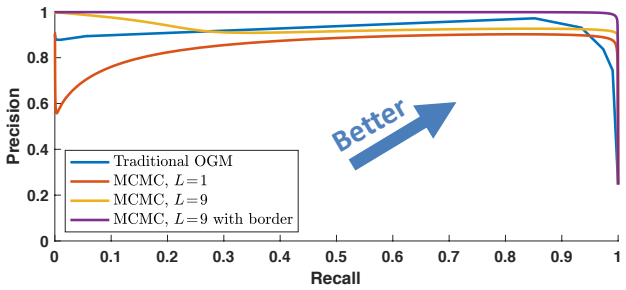


Fig. 5: Precision-Recall curve for simulation results, where each MCMC algorithm drew over 100 map samples. The MCMC algorithm with a border-based data-driven patch prior outperforms the other algorithms for all thresholds.

all unmapped cells are estimated at the cell prior. Figure 5 shows that traditional OGM can have higher precision than the MCMC algorithms that do not incorporate a patch border. In fact, the plot is somewhat deceiving as both MCMC algorithms have a higher maximum F_1 score as shown in Table II, but they can do worse on precision for very high threshold values (i.e., > 0.97) as traditional OGM can be overconfident in the marginal cell occupancy probability. Overall, the MCMC algorithm with a border-based data-driven patch prior outperforms the other algorithms on all measures. This result is highlighted in Table II and by the fact that it dominates the other algorithms in Figure 5.

The MCMC algorithms with a patch prior were also compared to traditional OGM on two hardware datasets: the Intel Research Lab [13] and the MIT Stata Center [8]. Both datasets contain dense laser range measurements of an indoor environment taken from a mobile robot. The dense laser data is useful for estimating the robot's pose and creating a benchmark map. The dense data is then subsampled to generate sparse data for the experiments, similar to [27]. This approach was not required in simulation because the robot pose and ground truth map were known.

The benchmark map is created using traditional OGM with all of the data, then thresholding each cell in the map at an occupancy probability of 0.20 because, as shown in Table II, this value yields a high F_1 score for traditional OGM. The hardware benchmark maps are shown in Figure 9, where occupied cells are shown in black and the remaining cells are unoccupied. The benchmark maps are overlaid with green, red, and blue rays that represent the first 10 cells mapped by each measurement ray in the three experiments conducted on each dataset. Table III details the three experiments conducted on each dataset and the

TABLE III: Experiments conducted on the hardware datasets. The Intel Lab map measures 30m×30m with a grid resolution of 2.0cm; the benchmark has a 1.0° increment and each experiment has a 2.0° increment between measurements. The MIT Stata map measures $50\text{m} \times 50\text{m}$ with a grid resolution of 3.3cm; the benchmark has a 0.25° increment and each experiment has a 2.5° increment between measurements. Both maps have $K = 2.25 \times 10^6$ cells.

	Number of Poses	Total Measurements	Portion of Cells Mapped
Intel Benchmark	910	163 800	59.63%
Experiment 1	910	81 900	55.78%
Experiment 2	455	40 950	50.27%
Experiment 3	91	8190	28.22%
MIT Benchmark	1977	2 056 080	16.21%
Experiment 4	1977	205 608	14.58%
Experiment 5	198	20 592	11.74%
Experiment 6	20	2080	5.23%

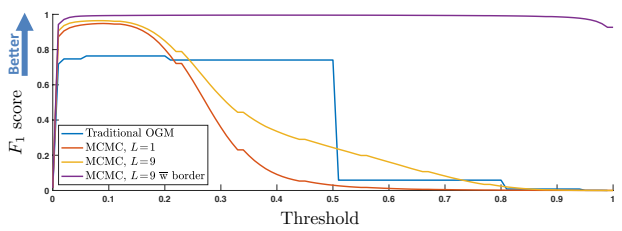


Fig. 6: F_1 score for simulation results as a function of cell threshold value. The MCMC algorithm with a border-based data-driven patch prior is very robust to the threshold value as it has a high F_1 score for nearly all cell threshold values.

proportion of range measurements used for each. The first experiment on each dataset uses the green, red, and blue rays; the second experiment uses the red and blue rays; the third experiment uses only the blue measurement rays. Therefore, each subsequent experiment uses increasingly sparse data.

Table IV summarizes the computation time for the MCMC algorithms on the hardware datasets and reinforces the results from Table I that the border-based method is faster. In computing the F_1 score for the hardware results, certain unmapped cells are omitted from the computation because the bounding box chosen can affect the result. Specifically, only those within 10 cells of a cell mapped by a range measurement are used to compute the F_1 score. This includes 73.04% of cells for the Intel Lab dataset and 24.09% for the MIT Stata dataset. Additional details are available in [21].

The hardware results are summarized in Figure 7 and Figure 8 by plotting the F_1 score of each OGM algorithm against a decreasing number of range measurements (i.e., each experiment). Both plots show that all algorithms do well on the first experiment and that traditional OGM has the highest F_1 score - this is expected as traditional OGM was used to generate the benchmark. The second experiment on each dataset shows that as sparsity increases, the MCMC algorithm with a border-based data-driven prior outperforms the others. The third experiment shows that this algorithm outperforms the others by a significant margin when data is very sparse. Figure 9 illustrates this finding by showing the

resulting OG map using this algorithm and comparing it to traditional OG map. By correlating neighbouring cells in the prior, the MCMC algorithm generates an OG map that better estimates the benchmark and could potentially be used for navigation, localization, and other tasks.

V. Conclusion and Future Work

This paper has developed an anytime algorithm that can improve an OG estimate by incorporating a data-driven patch prior into the MCMC OGM method. The method is shown to produce maps that better estimate a benchmark map, particularly when range measurement data is sparse. As robots move faster, measure in 3D, and use range sensors that measure farther, the data will become increasingly sparse and this method will yield more dramatic improvements.

Future work on this topic should explore correlating cells in only the prior (i.e., not measurements) and the effect of conditioning the prior on fewer neighbouring cells. Also, this work should be tested on more diverse maps including extending it to 3D data to highlight its effectiveness on increasingly sparse data.

Acknowledgment

The authors thank David Yoon from UTIAS for his contribution to this work and the Natural Sciences and Engineering Research Council of Canada (NSERC) for their support.

TABLE IV: Average time to compute one MCMC map sample (K/L patches, $L = 9$) using unoptimized Matlab code for hardware experiments.

	No border [s]	With border [s]
Exp. 1	6433	**1518**
Exp. 2	3023	**723.6**
Exp. 3	306.1	**166.8**
Exp. 4	4158	**1734**
Exp. 5	557.2	**201.2**
Exp. 6	44.71	**33.14**

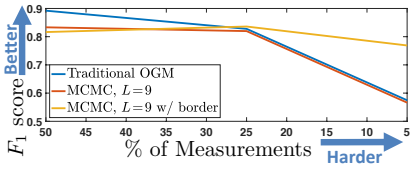


Fig. 7: F_1 scores for Intel Lab dataset experiments with 50%, 25% and 5% of measurements. All of the OGM algorithms do well when the area is densely mapped, but the MCMC algorithms that use a data-driven patch prior are able to better estimate the map with sparse measurements.

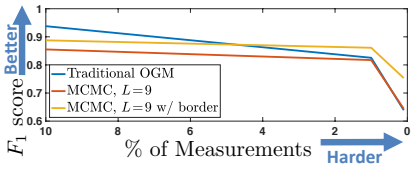


Fig. 8: F_1 scores for MIT Stata dataset experiments with 10%, 1% and 0.1% of measurements. All of the OGM algorithms do well when the area is densely mapped, but the MCMC algorithms that use a data-driven patch prior are able to better estimate the map with sparse measurements.

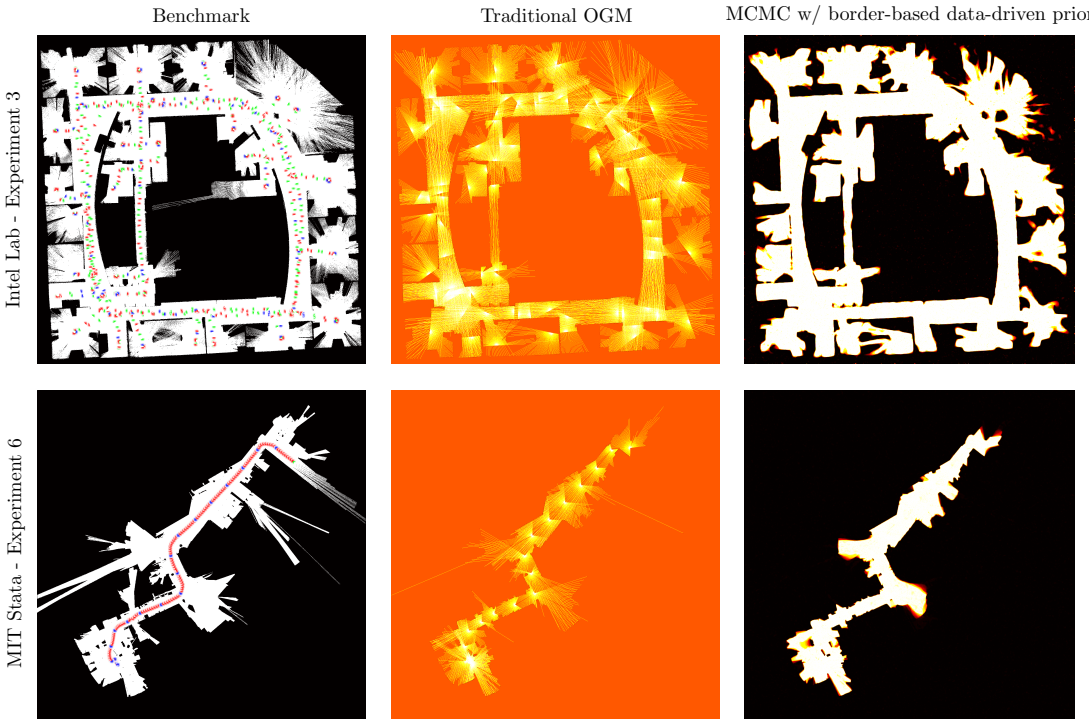


Fig. 9: Hardware results for Experiments 3 and 6. The benchmark maps are generated using all of the available range measurements. The two occupancy grids for each experiment shown are generated using a subset of the range measurements – only measurements shown in blue on the benchmark map. With a border-based, data-driven prior the MCMC algorithm is able to better estimate the benchmark map as compared to traditional OGM. Figure 1 shows the legend used.

References

[1] A.-A. Agha-Mohammadi, E. Heiden, K. Hausman, and G. Sukhatme, "Confidence-rich grid mapping," *The International Journal of Robotics Research*, pp. 1–23, 2019.

[2] D. Anguelov, D. Koller, E. Parker, and S. Thrun, "Detecting and modeling doors with mobile robots," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2004.

[3] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," *International conference on Machine learning (ICML)*, 2006.

[4] V. Dhiman, A. Kundu, F. Dellaert, and J. J. Corso, "Modern MAP inference methods for accurate and fast occupancy grid mapping on higher order factor graphs," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2037–2044, May 2014.

[5] V. Dhiman, "Towards better navigation: Optimizing algorithms for mapping, localization and planning," Ph.D. dissertation, University of Michigan, Horace H. Rackham School of Graduate Studies, 2019.

[6] K. Doherty, T. Shan, J. Wang, and B. Englot, "Learning-aided 3-D occupancy mapping with Bayesian generalized kernel inference," *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 953–966, Aug. 2019.

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

[8] M. Fallon, H. Johannsson, M. Kaess, and J. J. Leonard, "The mit stata center dataset," *The International Journal of Robotics Research*, vol. 32, no. 14, pp. 1695–1699, 2013. [Online]. Available: https://doi.org/10.1177/0278364913509035

[9] C. Georgiou, S. Anderson, and T. Dodd, "Constructing informative bayesian map priors: A multi-objective optimisation approach applied to indoor occupancy grid mapping," *The International Journal of Robotics Research*, vol. 36, no. 3, pp. 274–291, 2017.

[10] V. Guizilini and F. Ramos, "Learning to reconstruct 3d structures for occupancy mapping from depth and color information," *The International Journal of Robotics Research*, vol. 37, no. 13-14, pp. 1595–1609, 2018.

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

[12] L. Hou, X. Chen, K. Lan, R. Rasmussen, and J. Roberts, "Volumetric next best view by 3D Occupancy Mapping using Markov Chain Gibbs Sampler for precise manufacturing," *IEEE Access*, vol. 7, pp. 121 949–121 960, 2019.

[13] A. Howard and N. Roy, "The robotics data set repository (radish)," 2003. [Online]. Available: http://radish.sourceforge.net/

[14] M. G. Jadidi, J. V. Miro, and G. Dissanayake, "Gaussian processes autonomous mapping and exploration for range-sensing mobile robots," *Autonomous Robots*, no. 42, pp. 273–290, 2018.

[15] E. Kaufman, K. Takami, T. Lee, and Z. Ai, "Autonomous exploration with exact inverse sensor models," *Journal of Intelligent & Robotic Systems*, vol. 92, no. 3, pp. 435–452, Dec 2018.

[16] Y. Kwon, B. Moon, and S.-E. Yoon, "Adaptive kernel inference for dense and sharp occupancy grids," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2020, pp. 4712–4719.

[17] B. Lee, C. Zhang, Z. Huang, and D. D. Lee, "Online continuous mapping using gaussian process implicit surfaces," in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 6884–6890.

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

[19] ——, "Occupancy grid mapping with Markov Chain Monte Carlo Gibbs sampling," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3168–3174, May 2013.

[20] ——, "Optimizing online occupancy grid mapping to capture the residual uncertainty," *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[21] R. S. Merali, "A study of the benefits of cell correlations in occupancy grid mapping," Ph.D. dissertation, University of Toronto, 2020.

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

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

[24] M. Muffert, "Incremental map building with markov random fields and its evaluation," Ph.D. dissertation, University of Bonn, 2018.

[25] S. O'Callaghan, F. Ramos, and H. Durrant-Whyte, "Contextual occupancy maps using gaussian processes," in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, May 2009, pp. 1054 – 1060.

[26] S. T. O'Callaghan and F. T. Ramos, "Gaussian process occupancy maps," *I. J. Robotic Res.*, vol. 31, no. 1, pp. 42–62, 2012.

[27] C. O'Meadhra, W. Tabib, and N. Michael, "Variable resolution occupancy mapping using Gaussian mixture models," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2015–2022, Apr. 2019.

[28] F. Ramos and L. Ott, "Hilbert maps: Scalable continuous occupancy mapping with stochastic gradient descent," *The International Journal of Robotics Research*, vol. 35, no. 14, pp. 1717–1730, 2016.

[29] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[30] A. Schaefer, D. Buscher, L. Luft, and W. Burgard,

"A maximum likelihood approach to extract polylines from 2-D laser range scans," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2018.

[31] R. Senanayake and F. Ramos, "Bayesian hilbert maps for dynamic continuous occupancy mapping," in *Conference on Robot Learning*, 2017, pp. 458–471.

[32] K. S. Shankar and N. Michael, "MRFMap: Online probabilistic 3D mapping using forward ray sensor models," in *Robotics: Science and Systems*, Jul. 2020.

[33] R. Shrestha, F.-P. Tian, W. Feng, P. Tan, and R. Vaughan, "Learned map prediction for enhanced mobile robot exploration," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 1197–1204.

[34] 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.

[35] W. Zhi, L. Ott, R. Senanayake, and F. Ramos, "Continuous occupancy map fusion with fast bayesian hilbert maps," in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 4111–4117.