From Grids to Continuous Occupancy Maps through Area Kernels

Carlos E.O. Vido and Fabio Ramos

***Abstract*— In this work, we introduce a novel method for two-dimensional occupancy mapping using Gaussian processes. We address mapping as the task of classifying the robot's environment between free and occupied regions. The biggest challenge when using Gaussian processes for this task is the size of the input datasets. We tackle this problem by introducing a novel kernel, able to use as input data aggregated into two-dimensional cells. Using this kernel, we achieve comparable performance to previous Gaussian process occupancy mapping techniques in a fraction of the time taken by them. The approach can also be used to convert popular occupancy grids into continuous Gaussian process occupancy maps.**

I. Introduction

For any autonomous system, creating an accurate representation of its environment is an essential step to efficiently interact with it. As such, mapping plays a role of central importance in robotic navigation and path planning. One of the most popular method used for this task, occupancy grid maps (OGMs) was developed in the 80s by Moravec and Elfes [1]. The method is versatile, easy to implement and computationally efficient, which accounts for its widespread use, especially in 2D mapping.

OGMs, however, are not without shortcomings. Arguably, their most notorious deficiency comes from the standard approach of breaking down the task of mapping an area into the binary estimation of whether or not each grid cell is occupied. This approach makes the very strong assumption that the cells are independent, i.e., the occupancy of each cell is not affected by its neighbours. More than a mere approximation, this simplification ultimately ignores an important property of the very system it is trying to describe: obstacles in real world are physical entities with fairly regular, continuous structures.

As expected, this impacts the quality of the predictions made. Ignoring the spatial dependance between cells causes OGMs to have very high uncertainty in regions which are occluded, between data points or where readings are otherwise sparse. Using a model that does not exclude spatial correlations could help overcome this, resulting in more accurate and reliable maps.

A possible way to circumvent these problems is modelling the data using a method that is able to infer spatial correlations among data points. This creates an opportunity to use Gaussian processes (GPs), a Bayesian inference method that is very apt at nonlinear interpolation. To fit a nonparametric function through the data, GPs use a kernel which encodes a prior belief over the correlation between data points [2]. GP-based continuous occupancy mapping methods (GPOMs) have been proposed in the past [3], [4] and provide the basis for this work.

In a robotic mapping scenario, GPs face challenges of their own. They are not quick to compute, with a computational complexity that scales cubically with the number of inputs. To further this problem, the laser sensors commonly used in robotic environment sensing generate tens of thousands of readings every second.

The computational complexity of GPs is hard to tackle, since it stems from a very fundamental operation of inverting a Gram matrix built on the data points. This work proposes a technique that (1) preserves the strengths of GPOMs, but also (2) reduces the number of inputs by using kernels calculated over areas, thus condensing several data points into a single input for the GP. This last item is of great importance since, given the complexity of the task at hand, even a small reduction in the number of inputs can have a significant impact on performance.

II. Background

Robotic mapping can be interpreted [5] as the calculation of the map m as a posterior over the space of all possible maps, given the sensory information $\mathbf{z}$ collected by the robot in its path, which is taken to be a collection of poses $\mathbf{s}$:

$$p(m|\mathbf{z},\mathbf{s}). \tag{1}$$

The standard OGM approach is to partition m into a finite number of cells $\mathbf{m}_i$. To each of these, a binary occupancy value is attached, indicating whether or not that particular cell is occupied. Then the map is approximated by the product of all posterior probabilities:

$$p(m|\mathbf{z},\mathbf{s}) = \prod_i p(\mathbf{m}_i|\mathbf{z},\mathbf{s}). \tag{2}$$

This is where the weakness of the method lies, and several attempts have been made to address this shortcoming. An interesting approach using Bayesian inference is the GPOM [3], which tries to regress the occupancy value $\overline{y}$ in location $\mathbf{x}$ using a Gaussian Process prior with mean function $\mu(\mathbf{x})$ and covariance function $k(\mathbf{x},\mathbf{x}')$:

$$\overline{y}(\mathbf{x}) = \mathcal{GP}(\mu(\mathbf{x}), k(\mathbf{x},\mathbf{x}')). \tag{3}$$

Estimating the mean value for new query points requires computing the covariance matrix between all input $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ and query $\mathbf{X}_* = \{\mathbf{x}_*^j\}_{j=1}^M$ points, a noise term σ_n,

C.E.O. Vido and F.T. Ramos are with the School of Information Technologies, University of Sydney, NSW 2008, Australia. kadu@it.usyd.edu.au; fabio.ramos@sydney.edu.au

and the observations $\mathbf{y} = \{y_i\}_{i=1}^{N}$ associated with inputs $\mathbf{X}$:

$$\overline{\mathbf{y}}(\mathbf{X}_*) = K_*^\top (K + \sigma_n^2 I)^{-1} \mathbf{y}, \tag{4}$$

where, $\quad K = k(\mathbf{X}, \mathbf{X})$ and
$\quad K_* = k(\mathbf{X}, \mathbf{X}_*)$.

Here lies the reason behind the complexity of the algorithm. If we take the number of input points to be N and the test points to be M, this calculation alone has a complexity of $O(N^3 + N^2 M)$. Despite this challenge, the method is quite robust: it not only incorporates contextual information about the environment into the predictions—removing the independence assumption that plagues OGMs— but also returns a variance plot that can be used to create exploration strategies to improve the quality of the map.

Of special interest for the present work are previous GPOM methods that used integral kernels to measure relations between higher dimensional geometrical elements (like lines or areas) as well as points. This is known as *change of support* in the geostatistics community.

Within the realm of robotic mapping, methods using change of support have been proposed in the past [6][4]. They extend standard GP kernels by allowing the calculation of covariance matrices between points and lines ($K_{\mathbf{x}l}$) or sets of lines ($K_{ll'}$). This is achieved by integrating traditional kernels k:

$$K_{\mathbf{x}l} = \int_{\mathbf{x} \in l} k(\mathbf{x}, \mathbf{x}')\, d\mathbf{x}, \tag{5}$$

$$K_{ll'} = \int_{\mathbf{x}' \in l'} \int_{\mathbf{x} \in l} k(\mathbf{x}, \mathbf{x}')\, d\mathbf{x}\, d\mathbf{x}'. \tag{6}$$

These models are designed to be used with laser rangefinder sensory information, and the lines integrated over coincide with the laser beams—that is, they connect the positions from which the beams were emitted to where they were reflected. The results obtained were encouraging when compared to naïve GPOM and to OGM, as evidenced by comparing the receiver operator characteristic (ROC) curve for these methods (Figure 1 and Table I). The ROC curve is a plot of the false positive rate (FPR) versus the true positive rate (TPR) as the discrimination threshold varies, frequently used to illustrate the performance of a binary classifier.

TABLE I: GPOMIK ROC analysis [6]

Method	Accuracy	FPR when TPR = 90%
GPOMIK	0.9441	10.1%
GPOM	0.9162	79.57%
OGM	0.8938	21.9%

Another interesting use of change of support has been made in a different domain. In image processing, [7] uses an area kernel for image resolution enhancement. This kernel can represent covariances between points and areas (K_{xA}) or sets of areas ($K_{AA'}$), similar to equations 5 and 6 but using double integrals.

These are used to fuse images of different modalities (e.g. greyscale and colour) in different resolutions of the same subject. Each pixel in the lower resolution image is an input area A that corresponds to a set of pixels in the higher resolution image, each of which is on its turn an input point x. This allows for the discretisation of the problem, so K_{xA} and K_{AA} are given by:

$$K_{xA} = \frac{1}{N} \sum_{x \in A} k(\mathbf{x}, \mathbf{x}')\, d\mathbf{x}, \tag{7}$$

$$K_{AA'} = \frac{1}{NN'} \sum_{x' \in A'} \sum_{x \in A} k(\mathbf{x}, \mathbf{x}')\, d\mathbf{x}\, d\mathbf{x}', \tag{8}$$

where N and N' are the number of pixels x in a pixel A.

Change of support has been used in other kernel method applications. In anomaly detection, [8] uses kernels calculated over probability distributions $\mathbb{P}$ to generalise one-class support vector machines to a space of probability measures:

$$K(\widehat{\mathbb{P}}_1, \widehat{\mathbb{P}}_2) = \iint k(\mathbf{x}, \mathbf{x}')\, d\mathbb{P}_1(\mathbf{x})\, d\mathbb{P}_2(\mathbf{x}') \tag{9}$$

Rather than calculating anomalies in the data themselves, this approach allows the calculation of anomalies appearing as a result of the data's interactions, which the authors named *group anomalies* (as opposed to point anomalies). So instead of detecting groups of anomalous samples, for which there are other suitable detection algorithms, the method detects anomalous groups of normal samples, as illustrated in Figure 2. This kind of anomaly can only be observed in the space of distributions, so a kernel supported on distributions is better suited for the task.

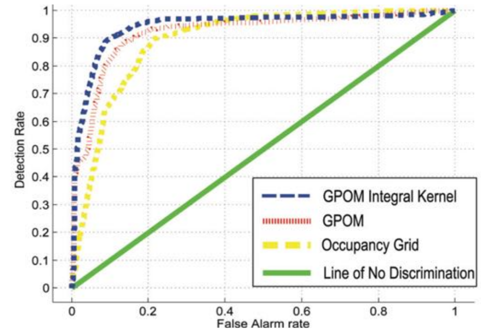


Fig. 1: ROC curve for GPOMIK, GPOM and OGM [6].

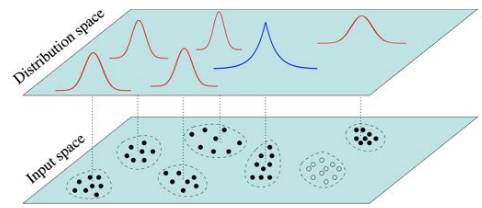


Fig. 2: Algorithms that detect groups of anomalous samples (unfilled points) are easy to detect on the input space. Kernels supported on distributions can be used to detect anomalous groups of normal samples (filled points) in the distribution space [8].

III. Occupancy Mapping with High-Dimensional Support Gaussian Processes

Our aim is to further extend kernels to extract information from arbitrarily defined areas. We can use such a tool to aggregate 2D laser rangefinder data into regions containing only free or only occupied points, effectively reducing the number of inputs to the Gaussian process. The output of the method is a continuous occupancy map, which can then be sampled in regular intervals to generate a grid map at any required resolution.

The algorithm here proposed is divided into three stages:

1) Input generation and pre processing:
   - collect data points from rangefinder;
   - separate data points between free (beams) and occupied (hits);
   - generate input cells.
2) Learning:
   - feed input cells into MSK-GP;
   - optimise kernel hyperparameters.
3) Post-processing and map generation:
   - generate occupancy probability surface;
   - probe at regular intervals to generate occupancy probability grid;
   - categorise grid points using thresholds to generate occupancy map.

Detailed descriptions of each stage are presented below.

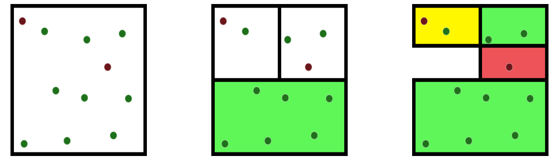
A. Input generation and pre-processing

The data generated by rangefinders usually consists of a set of distances from the robot's position to where each beam was reflected (a "hit"), indicating the position of obstacles within the range of the sensor. Another piece of information is available, though it might not be immediately obvious: that the region between the robot and the hits is empty. To fully incorporate this knowledge, we start by sampling each beam in regular distances, creating a set of free points.

To create the areas, first a rectangular cell orthogonal to the cartesian axes is put around the whole dataset. It is then divided in a process similar to a quad-tree, always along the longest axis, with two stopping criteria: once a cell has only free or only occupied points or when it is smaller than a size threshold, it is no longer divided. This process is illustrated in Figure 3. In regions with very wide uncluttered or unprobed areas (relative to the desired map resolution), a maximum size threshold can also be established, to ensure that the free cells do not become overly large, which could obscure a possible lack of measurements in the corresponding area.

B. Multi-support Gaussian process

In order to handle the input created in the pre-processing step, a GP formulated on areas is necessary. A fully functional kernel for this task must be able to handle mixed inputs containing points and areas. For this, we must create a pair of functions analogous to Equations 5 and 6, to calculate the covariance matrices between points and areas (K_{pa}) or pairs of areas ($K_{AA'}$):



(a) (b) (c)

Fig. 3: Cell generation process. Green points and areas are free, red are occupied. (a) All the points are put in an initial rectangular cell (0.9 × 1.0). (b) After the initial division, the bottom cell only has free points, so it is not further divided; upper element is divided again. (c) The top right cell is further divided into a free and an occupied element, while the top left is divided into an ambiguous area (in yellow) and an empty one, which is discarded.

$$K_{pA} = \iint_A k(\mathbf{x}, \mathbf{x}')\, d\mathbf{x}, \tag{10}$$

$$K_{AA'} = \iint_{A'} \iint_A k(\mathbf{x}, \mathbf{x}')\, d\mathbf{x}\, d\mathbf{x}'. \tag{11}$$

In the above equations, $k(\mathbf{x}, \mathbf{x}')$ can be replaced by virtually any positive semi-definite (PSD) kernel function. In spite of this, calculating the true value of these integrals can prove challenging even for simple kernels. An approximation is thus made: for a finite number of points $x \in A$,

$$K_{pA} \sim \sum_{\mathbf{x} \in A} k(\mathbf{x}, \mathbf{x}'), \tag{12}$$

$$K_{AA'} \sim \sum_{\mathbf{x}' \in A'} \sum_{\mathbf{x} \in A} k(\mathbf{x}, \mathbf{x}'). \tag{13}$$

Similar approximations have been used to construct integral kernels in other works [8], [9], since kernels constructed by direct summation of PSD kernels ware PSD themselves [10]. Moreover, even though the equations above were formulated for 2D cells, they could in theory be used for geometrical structures with any number of dimensions, which renders this result more flexible than directly or numerically solving Equations 10 and 11, as described in [4] and [6].

Once K_{pA} and $K_{AA'}$ are obtained, the covariance between two input vectors, each containing both points and areas is given by:

$$\mathbf{x} = \begin{bmatrix} A \\ \mathbf{p} \end{bmatrix},\ \mathbf{x}' = \begin{bmatrix} A' \\ \mathbf{p}' \end{bmatrix},\ K = \begin{bmatrix} k(A, A') & k(A, \mathbf{p}') \\ k(\mathbf{p}, A') & k(\mathbf{p}, \mathbf{p}') \end{bmatrix}. \tag{14}$$

In this final form, the kernel can be used to create a multi-support GP with the ability to handle only points, only areas or mixed input sets containing both. As we will show in the next section, however, this property is not entirely necessary for the remainder of this work.

C. Map generation and post-processing

Once the input cells have been generated and a suitable covariance function chosen, the hyperparameters can be trained using an optimisation algorithm to minimise the log marginal likelihood. Once this is done, the GP will have learnt an unbound surface.

To obtain a valid occupancy probability surface we must constrain this output to $[0;1]$, for which end we utilise a sigmoid function of the form

$$\varsigma(\mu, v) = \Phi\left(\frac{(\alpha \cdot \mu + \beta)}{\sqrt{1+\alpha^2 \cdot v}}\right), \tag{15}$$

where μ is the unbound output, v is the variance, Φ is the normal cumulative distribution function and α and β are parameters. This surface can be probed in regular intervals to yield a traditional grid map of any desired resolution, with real values for each cell.

To obtain a ternary output such as the one given by traditional OGMs, a pair of thresholds can be chosen, such that cells with probability of occupancy below the lower threshold can be set as free, those above the higher one can be set as occupied, and those in between as uncertain.

IV. Experiments

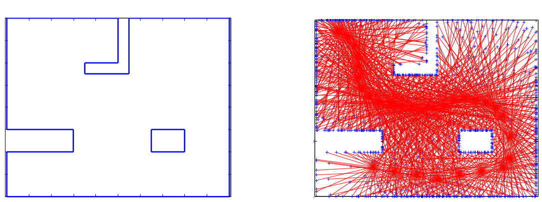
All the experiments described in this chapter are performed in a computer with a 3.2GHz processor and 8GB RAM. Accuracy measurements displayed represent the area under the receiver operating characteristic (ROC) curve. The false positive rate (FPR) for a fixed true positive rate (TPR) of 95% is offered as an additional performance metric.

Two different optimisers have been compared, the Broyden-Fletcher-Goldfarb-Shanno quasi-Newton method (BFGS) [11], [12], [13], [14] and simulated annealing [15], [16]. Since their impact in the accuracy of the method was negligible and this work does not focus on the optimisation step, comparative results were omitted.

The proposed method (MSK-GPOM) is benchmarked against the previous GPOM method using integral kernels (GPOMIK) described in [6], which has been demonstrated to outperform both OGMs and the GPOM method described in [3], as previously shown in Figure 1 and Table I.

A. Synthetic data

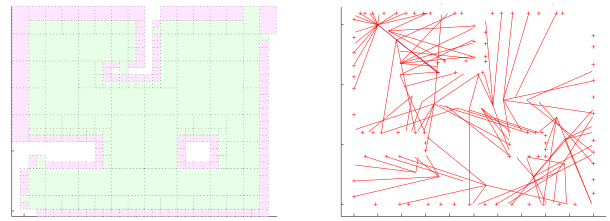
Initial tests used synthetic data for which the ground truth is known. It simulates a robot equipped with a laser rangefinder taking 36 noiseless readings over 360 degrees on each of 31 poses. It moves within a room roughly 20×16 arbitrary units of distance. Figure 4 has a visual representation of the ground truth and the dataset generated.



(a) (b)

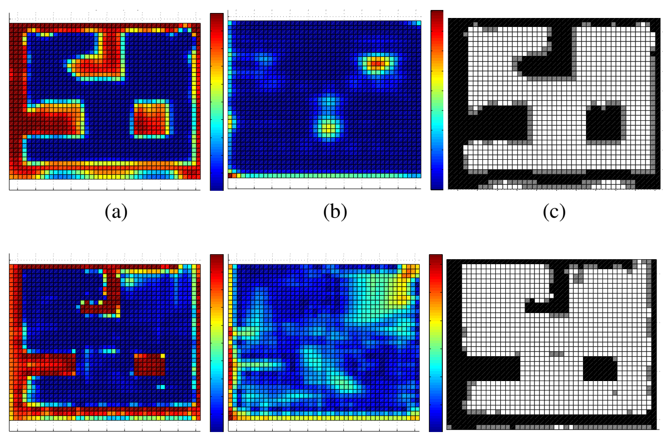
Fig. 4: Synthetic dataset. (a) Ground truth, walls represented in blue; (b) simulated dataset, red lines represent beams and blue crosses, hits.

Although the dataset may appear dense at a glance, state-of-the-art rangefinders can take readings every 0.5°, generating sets 20 times denser than this. GPOMIK deals with this through a routine that pre-selects a subset of relevant readings, which may vary according to the kernel being used. MSK-GPOM takes evenly spaced readings to avoid the computational cost of calculating which readings are relevant. The cells generated depend on the thresholds chosen, but not on the kernel. The input datasets for each of the methods compared, resulting from the different pre-processing steps, is shown in Figure 5.



(a) (b)

Fig. 5: Inputs for the compared methods generated from the synthetic dataset. (a) MSK-GPOM input cells using a threshold of 0.5, (b) GPOMIK input beams and hits using a square exponential kernel.



(a) (b) (c)

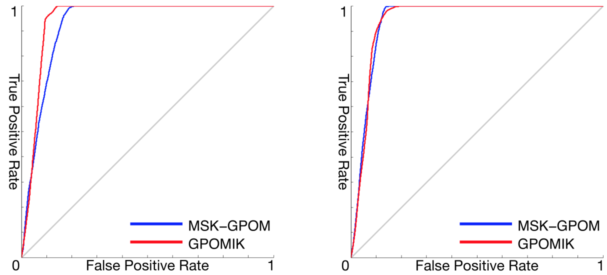
(d) (e) (f)

Fig. 6: Outputs for the compared methods using the synthetic dataset, MSK-GPOM on top and GPOMIK below it. (a) and (d) Probability of occupancy, blue means zero, red means one; (b) and (e) predictive variance, blue is low, red is high; (c) and (f) Occupancy map, white is free, black is occupied, grey is unknown.

TABLE II: Benchmark times on the synthetic dataset

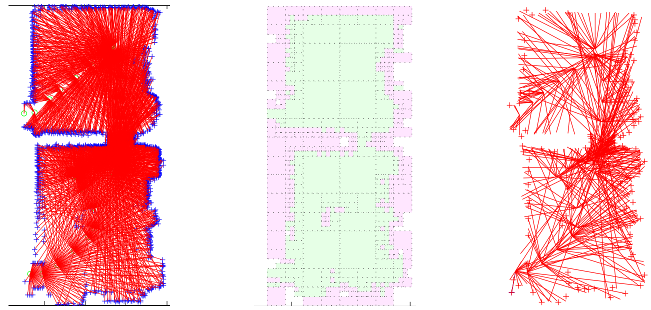
Method	Kernel	Running time	Accuracy	FPR when TPR = 95%
GPOMIK	Matérn 3	79.46s	0.9447	12.25%
	Sq. Exp.	86.58s	0.9415	9.76%
MSK-GPOM	Matérn 3	23.76s	0.9416	12.34%
	Sq. Exp.	19.33s	0.9266	16.30%

After feeding the inputs to a GP, both methods generate the same types of outputs: a continuous probability of occupancy



(a) (b)

Fig. 7: ROC curves for the compared methods in the synthetic data benchmark. GPOMIK in red, MSK-GPOM in blue, no discrimination line in grey. (a) Square exponential kernel; (b) Matérn 3 kernel.



(a) (b) (c)

Fig. 8: Real dataset. (a) Subset of the data used, red lines represent beams, blue crosses represent hits, green circles represent the robot's positions in each frame; (b) input for MSK-GPOM; (c) input for GPOMIK.

surface and a continuous predictive variance surface. Then we can sample these surfaces in the desired resolutions to generate a map. If desired, thresholds can be used to constrain the map to the three states in a traditional occupancy grid map—namely "free", "occupied" and "unknown". In Figure 6 we can see the output maps, sampled using a grid of squares of side 0.5. To generate the three-state map, cells with occupancy probability above 66% were considered occupied, those below 33%, free and the remainder, unknown.

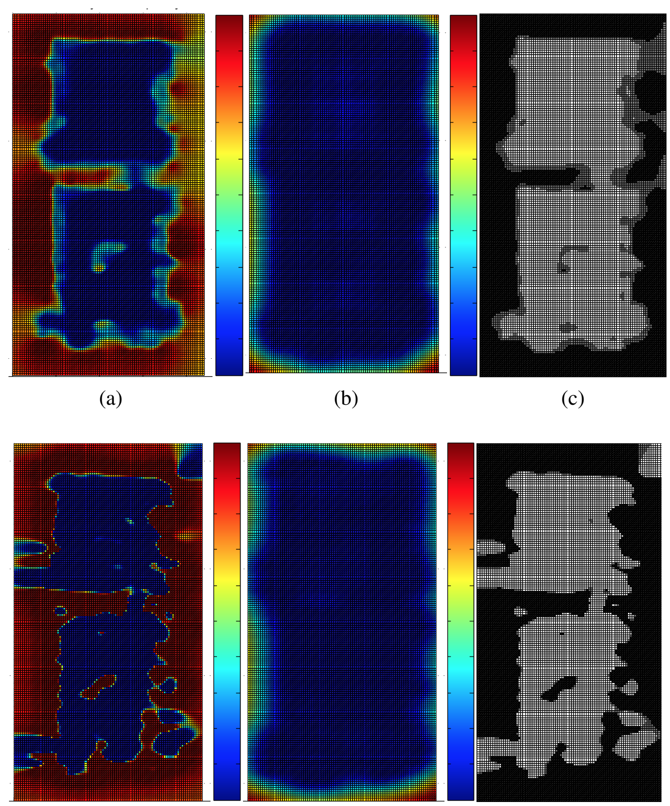
In Table II and Figure 7 we can see a numerical comparison between both methods. Results are displayed for GPs using both the square exponential kernel (Figures 5 and 6) and the Matérn 3 kernel. For this synthetic dataset, we can verify that while running in one quarter of the time taken by GPOMIK, MSK-GPOM achieved comparable accuracy.

B. Real data

Although the benchmarks in the previous subsection suggest the algorithm achieves performance comparable to GPOMIK's, synthetic data offers a lot of conveniences that in a real setting would not exist, such as lack of sensor noise. In order to properly evaluate the MSK-GPOM algorithm, we have tested it using datasets taken by actual robots.

The dataset used was provided by Dirk Haehnel, and contains 395 poses with 361 equally spaced laser readings spanning 180 degrees taken from each. The collection site is Belgioioso Castle, located in Milan. The dataset is made available on the Robotics Datasets webpage [17], maintained by Cyrill Stachniss. The data are presented both in raw form and after loop closure; the latter was used in this benchmark. A subset of 52 poses was used, from each of which 37 equally spaced laser beams (out of 361) were taken into consideration. For the ground truth, a different subset of the data was used, with a balanced amount of free and occupied points. These data refer to two adjacent rooms in the castle, and can be seen in Figure 8, along with the input sets generated for each method. In these tests, inputs and results shown are generated using the Matérn 3 kernel.

The outputs generated from the real dataset are shown in Figure 9. The thresholds for the occupancy maps are the same as in the previous experiment, but the sampling grid used is denser, generating a higher resolution map.



(a) (b) (c)

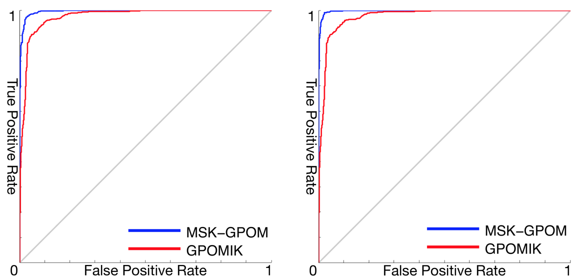
(d) (e) (f)

Fig. 9: Outputs for the compared methods, MSK-GPOM on top and GPOMIK below it. (a) and (d) Probability of occupancy; blue means zero, red means one; (b) and (e) predictive variance; blue is low, red is high; (c) and (f) Occupancy map; white is free, black is occupied, grey is unknown.

In Table III and Figure 10 we can see a numerical comparison between both methods. The Belgioioso dataset is larger than the synthetic one, and the grid used is finer,

TABLE III: Benchmark times on the real-world dataset

Method	Kernel	Running time	Accuracy	FPR when TPR = 95%
GPOMIK	Matérn 3	445.42s	0.9751	9.60%
	Sq. Exp.	607.93s	0.9665	8.20%
MSK-GPOM	Matérn 3	52.07s	0.9918	3.30%
	Sq. Exp.	37.29s	0.9947	2.00%



(a) (b)

Fig. 10: ROC curves for the compared methods in the real data benchmark. GPOMIK in red, MSK-GPOM in blue, no discrimination line in grey. (a) Square exponential kernel; (b) Matérn 3 kernel.

making the difference between both methods is more evident. MSK-GPOM achieved a higher accuracy in a fraction of the time taken by GPOMIK.

V. Conclusions and Future Work

The contributions presented in this work are twofold. Firstly, we introduce a multi-support kernel that is easy to implement and enables traditional covariance functions to accept as input not only points, but also two-dimensional regions. This kernel can be used to reduce the size of covariance matrices, accelerating Gaussian process inference and learning. Then, we elaborate a continuous occupancy mapping technique using a GP with the aforementioned kernel to handle uncertainty. It demonstrated comparable accuracy in relation to similar state-of-the-art techniques, while taking a much smaller toll on speed when handling large datasets.

These are only initial results. Even though the tests performed were all for two-dimensional datasets, the kernel presented can work in three dimensions as well. This means the method could be tested using three dimensional datasets after some small changes. The kernel developed is not tailored for the problem at hand, so it could also prove fruitful to investigate whether it can be used in other GP applications to handle high-dimensional data.

The method still needs to face some robotic mapping challenges. Currently, it does not support online learning, which hinders its deployment in real-world full automation scenarios. Furthermore, it was designed to use inputs from laser rangefinders, which are not always available due to the equipment's cost. It would become more versatile if it could be adapted to use other data modalities or perform data fusion. Lastly, it has not been designed to take advantage of parallel computer architectures that would render it even faster, which could increase its appeal.

This method also offers opportunities to solve other applications. Since it takes arbitrary two-dimensional inputs, it could potentially use occupancy maps stored as grids or quad-trees as input for the GP. This would provide a convenient method for redefining the resolution of existing discrete maps, as well as converting them to continuous maps. A three-dimensional implementation could do the same with maps stored as oct-trees, that are very popular in three-dimensional mapping.

VI. Acknowledgements

This work was partially supported by NICTA. NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

References

[1] H. Moravek and A. Elfes, "High resolution maps from wide angle sonar," in *Proceedings of the IEEE International Conference on Robotics and Automation*, March 1985,, pp. 116–121.

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

[3] S. T. O'Callaghan and F. T. Ramos, "Gaussian process occupancy maps," *The International Journal of Robotics Research*, 2012.

[4] S. Kim and J. Kim, "Continuous occupancy maps using overlapping local gaussian processes," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, November 2013.

[5] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.

[6] S. T. O'Callaghan and F. T. Ramos, "Continuous occupancy mapping with integral kernels," in *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011, pp. 1494–1500.

[7] A. Reid, F. Ramos, and S. Sukkarieh, "Bayesian fusion for multimodal aerial images," in *Proceedings of Robotics: Science and Systems*, Berlin, Germany, June 2013.

[8] K. Muandet and B. Schölkopf, *One-class support measure machines for group anomaly detection*. arXiv:1303.0309 [stat.ML], 2013.

[9] A. E. Gelfand, L. Zhu, and B. P. Carlin, "On the change of support problem for spatio-temporal data," *Biostatistics*, vol. 2, no. 1, pp. 31–45, 2001.

[10] T. J. Hastie and R. J. Tibshirani, *Generalized Additive Models*. Chapman and Hall, 1990.

[11] C. G. Broyden, "The convergence of a class of double-rank minimization algorithms," *Journal of the Institute of Mathematics and Its Applications*, vol. 6, pp. 76–90, 1970.

[12] R. Fletcher, "A new approach to variable metric algorithms," *Computer Journal*, vol. 13, no. 3, pp. 317—322, 1970.

[13] D. Goldfarb, "A family of variable metric updates derived by variational means," *Mathematics of Computation*, vol. 24, no. 109, pp. 23—26, 1970.

[14] D. F. Shanno, "Conditioning of quasi-newton methods for function minimization," *Mathematics of Computing*, vol. 24, no. 111, pp. 647–656, 1970.

[15] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *SCIENCE*, vol. 220, no. 4598, pp. 671–680, 1983.

[16] V. Černý, "Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm," *Journal of Optimization Theory and Applications*, vol. 45, pp. 41–51, 1985.

[17] C. Stachniss, "Robotics datasets." [Online]. Available: http://www2.informatik.uni-freiburg.de/ stachnis/datasets/