Proceedings of Australasian Conference on Robotics and Automation, 3-5 Dec 2012, Victoria University of Wellington, New Zealand.

Building Large-Scale Occupancy Maps using an Infinite Mixture of Gaussian Process Experts

Soohwan Kim and Jonghyuk Kim
The Australian National University, Australia
{soohwan.kim, jonghyuk.kim}@anu.edu.au

Abstract

This paper proposes a novel method of occupancy map building for large-scale applications. Although Gaussian processes have been successfully applied to occupancy map building, it suffers from high computational complexity of $O(n^3)$, where n is the number of training data, limiting its use for large-scale mappings. We propose to take a divide-and-conquer approach by partitioning training data into manageable subsets by combining a Dirichlet process mixture on top of a Gaussian process, which turns into an infinite mixtures of Gaussian process experts. Experimental results with simulated data show that our method produces accurate occupancy maps while maintaining the scalability.

1 Introduction

Mapping is one of the fundamental problems for mobile robots to understand surrounding environments to fulfil dependable tasks such as navigation or interaction with the environments.

Among various map representations, *occupancy grid maps* have been widely used since it is both fast and accurate. [7, 15] However, discretising the space into grid cells requires a lot of memory, which makes it inapplicable to large-scale environments. This problem was addressed by Wurm *et al.* [17] by applying a hierarchical data structure called *octomap*.

In addition, the independency assumption between grid cells which helps occupancy grid maps be updated fast makes the results sparse and inconsistent. Thrun [15] relaxed this strict assumption with *forward sensor models* and estimated consistent occupancy grid maps with sonar sensors, but the *expectation-maximisation* technique suffered from high computational time.

Recently, Lang *et al.* [6] have applied *Gaussian process regression* [14] with non-stationary covariance functions to build *elevation maps* with terrain data. Hadsell *et al.* [2] have extended this approach by exploiting visibility information to constrain the terrain surface and enhanced the accuracy. Elevation maps, however, are 2.5D and thus have difficulties in discriminating vertically overlapping objects such as tunnels and bridges.

Meanwhile, O'Callaghan *et al.* [11, 10] viewed occupancy map building as a binary classification problem and applied *Gaussian process classification*, which allows continuous input spaces and thus, needs no fixed map resolution in advance. Thanks to the dependency assumption between occupancy values, accurate and dense occupancy maps were obtained. However, the computational complexity of a Gaussian process grows in $O(n^3)$, where n is the number of training data. Thus, it is not directly applicable to large-scale environments.

Kim and Kim [4, 5] took a divide-and-conquer strategy to make it more scalable. They clustered training data to reduce the data size and applied individual Gaussian processes to each cluster to build local occupancy maps which were merged with a *mixture of experts* scheme [16]. With partitioned training data, the computational time was dramatically reduced, but clustering errors degraded the accuracy of the map.

In this paper, we further elaborate our previous work which has two separate steps of divide and conquer by optimally integrating them in one framework, an *infinite mixture of Gaussian process experts* [13]. Our method is scalable because it divides training data into subsets and also accurate because soft data clustering avoids possible clustering errors. With simulated data we will compare the performance of our method with the previous approaches and show that our method is accurate while maintaining the scalability.

The structure of the paper is as follows. In Section 2 we introduce an infinite mixture of Gaussian process experts. In Section 3 we propose our occupancy map building algorithm. Experimental results and comparison of accuracy and scalability with the previous work follow in Section 4. We conclude the paper in Section 5.

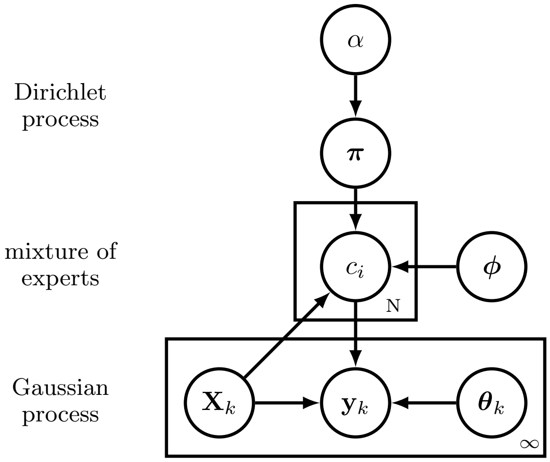


Figure 1: Graphical model of an infinite mixture of Gaussian process experts. Inputs $\mathbf{x}$ and outputs y_i are assigned to countably infinite Gaussian process experts with hyperparameters $\boldsymbol{\theta}_k$. The allocations c_i is determined by a mixture of experts with a gating network $\boldsymbol{\phi}$ and a Dirichlet process with a mixing proportion $\boldsymbol{\pi}$ and a concentration parameter α.

2 Infinite Mixture of Gaussian Process Experts

Fig. 1 shows the graphical model of an infinite mixture of Gaussian process experts [13]. From the bottom to the top, each component will be described in the following.

2.1 Gaussian Process

A Gaussian process [14] which is a Bayesian nonparametric approach to regression and classification is placed at the lower part of Fig. 1. It is defined with a mean function $m(\mathbf{x})$ and a covariance function $k(\mathbf{x}, \mathbf{x}')$,

$$f(\mathbf{x}) \sim \mathcal{GP}\left(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\right). \tag{1}$$

For noisy observations, we assume white noises,

$$y = f(\mathbf{x}) + \epsilon, \tag{2}$$

where $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$ and σ_n^2 is the noise variance.

From Eq. 1 and 2, given n noisy observations $(\mathbf{X}, \mathbf{y}) = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ and a test input $\mathbf{x}_*$, there exists a joint Gaussian distribution between the training outputs $\mathbf{y}$ and the test output f_*,

$$\begin{bmatrix} \mathbf{y} \\ f_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K} + \sigma_n^2 I & \mathbf{k}_* \\ \mathbf{k}_*^{\mathrm{T}} & k_{**} \end{bmatrix}\right), \tag{3}$$

where $\mathbf{K} \in \mathbb{R}^{n \times n}$, $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, $\mathbf{k}_* \in \mathbb{R}^n$, $\mathbf{k}_{*i} = k(\mathbf{x}_i, \mathbf{x}_*)$, and $k_{**} \in \mathbb{R}$, $k_{**} = k(\mathbf{x}_*, \mathbf{x}_*)$. Note that here the zero mean function is selected to simplify the case.

Therefore, the conditional distribution of the test output f_* is also a Gaussian distribution,

$$f_* | \mathbf{x}_*, \mathbf{X}, \mathbf{y} \sim \mathcal{N}\left(\mu_{f_*}, \sigma_{f_*}^2\right), \tag{4}$$

with a mean μ_{f_*} and a variance $\sigma_{f_*}^2$,

$$\begin{aligned} \mu_{f_*} &= \mathbf{k}_*^{\mathrm{T}} [\mathbf{K} + \sigma_n^2 \mathbf{I}]^{-1} \mathbf{y} \\ \sigma_{f_*}^2 &= k_{**} - \mathbf{k}_*^{\mathrm{T}} [\mathbf{K} + \sigma_n^2 \mathbf{I}]^{-1} \mathbf{k}_*. \end{aligned} \tag{5}$$

Covariance Function

In this paper, we choose the *Mátern* covariance function with $\nu = 3/2$,

$$k_{\nu=3/2}(r) = \left(1 + \frac{\sqrt{3}r}{l}\right) \exp\left(\frac{-\sqrt{3}r}{l}\right), \tag{6}$$

where $r = |\mathbf{x} - \mathbf{x}'|$, and the parameter l is non-negative.

This is because the squared exponential covariance function in the previous approaches [4, 9] is too smooth to model abrupt changes in occupancy values of arbitrary environments. Note that the *Mátern* covariance function is a generalisation of the squared exponential which is the special case when $\nu \to \infty$.

The hyperparameters $\boldsymbol{\theta}$ in Fig. 1 are defined as a set of the noise variance and the parameters of the covariance function, i.e. $\boldsymbol{\theta} = \{\sigma_n^2, l\}$ in our case.

Integral Kernels

Originally, a covariance function defines the similarity between two data points. Therefore, we need to discretise the laser beam segments into several knot points, which makes the training data even larger.

To avoid this problem, we apply *integral kernels* [9] which integrate the covariance function for the line-to-point covariance function $k_{lp}(\mathbf{l}, \mathbf{x})$ and double integrate for the line-to-line covariance function $k_{ll}(\mathbf{l}, \mathbf{l}')$,

$$\begin{aligned} k_{lp}(\mathbf{l}, \mathbf{x}) &= \int_0^1 k(\mathbf{l}(u), \mathbf{x}) du, \\ k_{ll}(\mathbf{l}, \mathbf{l}') &= \int_0^1 \int_0^1 k(\mathbf{l}(u), \mathbf{l}'(v)) du\, dv, \end{aligned} \tag{7}$$

where $\mathbf{l}(u)$ and $\mathbf{l}'(v)$ are line segments parameterised with $u, v \in [0, 1]$, respectively.

Since with an integral kernel the output y represents the net area under the function f, we multiply the training output $y = -1$ (empty) of a laser beam segment by its length.

Note that with the *Mátern* covariance function of Eq. 6, there exists no closed form solution for Eq. 7. In this paper, we employ *Clenshaw-Curtis quadrature* [1] to integrate covariance functions numerically as in [9].

Probabilistic Least-Squares Classification

For a occupancy mapping problem where the test output y_* is labelled as 1 (occupied) or -1 (empty), we apply *probabilistic least-squares classification* [12] to obtain the class probabilities between 0 and 1 by squashing the test output f_* through a cumulative Gaussian density function Φ,

$$p(y_* = 1|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \Phi\left(\frac{f_*(\alpha\mu_{f_*} + \beta)}{\sqrt{1 + \alpha^2\sigma^2_{f_*}}}\right), \quad (8)$$

where the parameters α and β are optimised by performing leave-one-out cross-validation on the training set.

2.2 Modified Mixture of Experts

In order to reduce the computational time of a Gaussian process, we partition the training data and assign individual Gaussian processes to each cluster by adding a mixture of experts at the centre of Fig. 1.

The original mixture of experts assumes that data points are i.i.d. given parameters $\boldsymbol{\theta}$ of K experts. However, this assumption does not apply to a Gaussian process where there exists a joint distribution between outputs. Instead, we sum over all assignments $\mathbf{c}$,

$$\begin{aligned} p(\mathbf{y}|\mathbf{X}, \boldsymbol{\phi}, \boldsymbol{\theta}) &= \sum_{\mathbf{c}} p(\mathbf{c}|\mathbf{X}, \boldsymbol{\phi}) p(\mathbf{y}|\mathbf{c}, \mathbf{X}, \boldsymbol{\theta}) \quad (9) \\ &= \sum_{\mathbf{c}} p(\mathbf{c}|\mathbf{X}, \boldsymbol{\phi}) \prod_{k=1}^{K} p(\{y_i : c_i = k\}|\{\mathbf{x}_i : c_i = k\}, \boldsymbol{\theta}_k). \end{aligned}$$

The gating network $p(\mathbf{c}|\mathbf{X}, \boldsymbol{\phi})$ for Gaussian process experts will be elaborated in the next subsection.

2.3 Input-Dependent Dirichlet Process

In the mixture of experts scheme, the number of clusters K and data allocation $\mathbf{c}$ are assumed to be given. That is why it suffers from the clustering error in our previous work. [4] Therefore, we add a Dirichlet process on top of the mixture of experts in Fig. 1 to tackle this problem.

The Dirichlet process mixture which consists of a Dirichlet process and a mixture of experts considers countably infinite number of experts which can be thought of as the limit of a mixture of experts as the number of experts K goes infinity.

Given the concentration parameter α, the probability of the mixing proportion $\boldsymbol{\pi} = \{\pi_k\}_{k=1}^K$ is assumed to be a symmetric Dirichlet distribution,

$$\begin{aligned} p(\pi_1, ..., \pi_K|\alpha) &= \text{Dirichlet}(\pi_1, ..., \pi_K; \alpha/K, ..., \alpha/K) \\ &= \frac{\Gamma(\alpha)}{\Gamma(\alpha/K)^K} \prod_{k=1}^{K} \pi_k^{\alpha/K-1}, \quad (10) \end{aligned}$$

where $\Gamma(z) = \int_0^\infty e^{-t}t^{z-1}dt$ is the gamma function.

On the other hand, the joint probability of indicator variables $\mathbf{c}$ given the mixing proportions $\boldsymbol{\pi}$ is assumed to be a multinomial distribution,

$$p(\mathbf{c}|\boldsymbol{\pi}) = \prod_{k=1}^{K} \pi_k^{n_k}, \quad (11)$$

where n_k is the number of data or *occupancy number* of cluster k such that $n_k = \sum_{i=1}^n \delta(c_i, k)$.

With Eq. 10, 11 and the integration of the standard Dirichlet distribution, the mixing proportion $\boldsymbol{\pi}$ can be integrated out as

$$p(\mathbf{c}|\alpha) = \frac{\Gamma(\alpha)}{\Gamma(n+\alpha)} \prod_{k=1}^{K} \frac{\Gamma(n_k + \alpha/K)}{\Gamma(\alpha/K)}. \quad (12)$$

By keeping all but the single indicator c_i fixed, we get

$$p(c_i = k|\mathbf{c}_{-i}, \alpha) = \frac{n_{-i,k} + \alpha/K}{n - 1 + \alpha}, \quad (13)$$

where $\mathbf{c}_{-i}$ represents all the indicators except c_i and $n_{-i,k}$ denotes the number of data of cluster k excluding i-th data.

Now, let the number of clusters K in Eq. 13 tend to infinity, we get

$$\begin{aligned} p(c_i = k|\mathbf{c}_{-i}, \alpha) &= \frac{n_{-i,k}}{n-1+\alpha}, \\ p(c_i \neq c_{i'}, \text{for all } i' \neq i|\mathbf{c}_{-i}, \alpha) &= \frac{\alpha}{n-1+\alpha}, \quad (14) \end{aligned}$$

where the second equation represents the probability of generating a new expert and makes the number of experts flexible and countably infinite. Note that the number of experts grows as the concentration parameter α increases.

To make the Dirichlet process input-dependent for the modified mixture of experts, the occupation number $n_{-i,k}$ is modified with a local estimate using a squared exponential $K_{\boldsymbol{\phi}}$ with a length-scale $\boldsymbol{\phi}$ as

$$n_{-i,k}(\mathbf{X}, \boldsymbol{\phi}) = (n-1)\frac{\sum_{i' \neq i} K_{\boldsymbol{\phi}}(\mathbf{x}_i, \mathbf{x}_{i'})\delta(c_{i'}, k)}{\sum_{i' \neq i} K_{\boldsymbol{\phi}}(\mathbf{x}_i, \mathbf{x}_{i'})}, \quad (15)$$

where $K_{\boldsymbol{\phi}}(\mathbf{x}_i, \mathbf{x}_{i'}) = \exp\left(-\frac{1}{2}\sum_d \frac{(x_{i,d} - x_{i',d})^2}{\phi_d^2}\right)$.

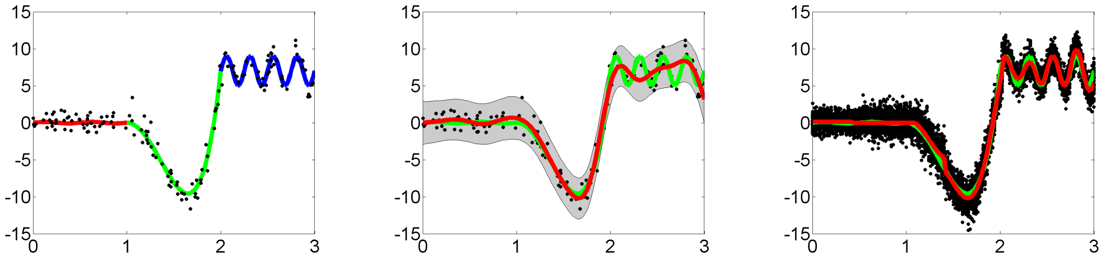
Now we can rewrite Eq. 14 as,

$$\begin{aligned} p(c_i = k|\mathbf{c}_{-i}, \mathbf{X}, \boldsymbol{\phi}, \alpha) &= \frac{n_{-i,k}(\mathbf{X}, \boldsymbol{\phi})}{n-1+\alpha}, \\ p(c_i \neq c_{i'}, \text{for all } i' \neq i|\mathbf{c}_{-i}, \mathbf{X}, \boldsymbol{\phi}, \alpha) &= \frac{\alpha}{n-1+\alpha}. \quad (16) \end{aligned}$$

2.4 Inference by Gibbs Sampling

The full joint distribution $p(\alpha, \boldsymbol{\pi}, \boldsymbol{\phi}, \mathbf{c}, \mathbf{X}, \mathbf{y}, \boldsymbol{\theta})$ of Fig. 1 is analytically intractable. Thus, in order to infer posterior distributions we apply Gibbs sampling [8].

Note that in this paper we do not put vague priors to α, $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ as in [13] because it delays the convergence of Gibbs sampling. Instead, we fix α and $\boldsymbol{\phi}$ and to 5 and 1s, respectively, and optimise $\boldsymbol{\theta}$ to minimise the marginal probabilities of each Gaussian process expert with its own training data. Therefore, we only need to draw samples on the indicator variables $\mathbf{c}$.



(a) Simulated data (b) Gaussian process (c) Infinite mixture of GP experts

Figure 2: Comparison between a Gaussian process and an infinite mixture of Gaussian processes. (a) The ground truth function consists of three distinct functions. (b) The Gaussian process (red) does not fit the ground truth (green) well in the right region. (c) The median (red) of samples (black dots) of infinite mixture of Gaussian process experts fits the ground truth (green) well in the whole region.

The conditional distribution of the indicator variables $\mathbf{c}$ given its *Markov blanket* is

$$\begin{aligned} & p(c_i = k|\mathbf{c}_{-i}, \mathbf{X}, \mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\pi}) \\ & \propto p(y_i|c_i = k, \mathbf{X}, \mathbf{y}_{-i}, \boldsymbol{\theta})\ p(c_i = k|\mathbf{c}_{-i}, \mathbf{X}, \boldsymbol{\phi}, \boldsymbol{\pi}), \end{aligned} \quad (17)$$

where $\mathbf{y}_{-i}$ denotes all the outputs $\mathbf{y}$ except y_i.

The first term on the right hand side in in Eq. 17 corresponds to the predictive probability of k-th Gaussian process experts with its own training data described as Eq. 4, while the second term is already derived with the mixing proportion $\boldsymbol{\pi}$ integrated out in Eq. 16 from a input-dependent Dirichlet process.

Note that the first term of Eq. 17 can be thought of as how suitable a data point is to a Gaussian process expert and the second term as how close it is to the expert's data set. Therefore, we can say that an infinite mixture of Gaussian process experts combines divide and conquer in one framework, where clustering is performed based on the similarity between data points and the matching score to each Gaussian process experts.

3 Occupancy Map Building Algorithm

The Algorithm 1 summarises the occupancy map building procedures using an infinite mixture of Gaussian process experts which was explained in the previous section.

In Line 1 and 2, we label laser hit points and laser beam segments as 1 (occupied) and −length (empty), respectively for the binary classification. In Line 3 we may start with one cluster which has all the training data or initially partition the training data to converge the Gibbs sampling faster. The readers may apply some clustering techniques such as k-means clustering in this step. From Line 6 to 10 we sample indicator variables $\mathbf{c}$ for clustering and predict a occupancy map with Gaussian processes of each cluster until it converge.

Algorithm 1 Occupancy Map Building using an Infinite Mixture of Gaussian Process Experts

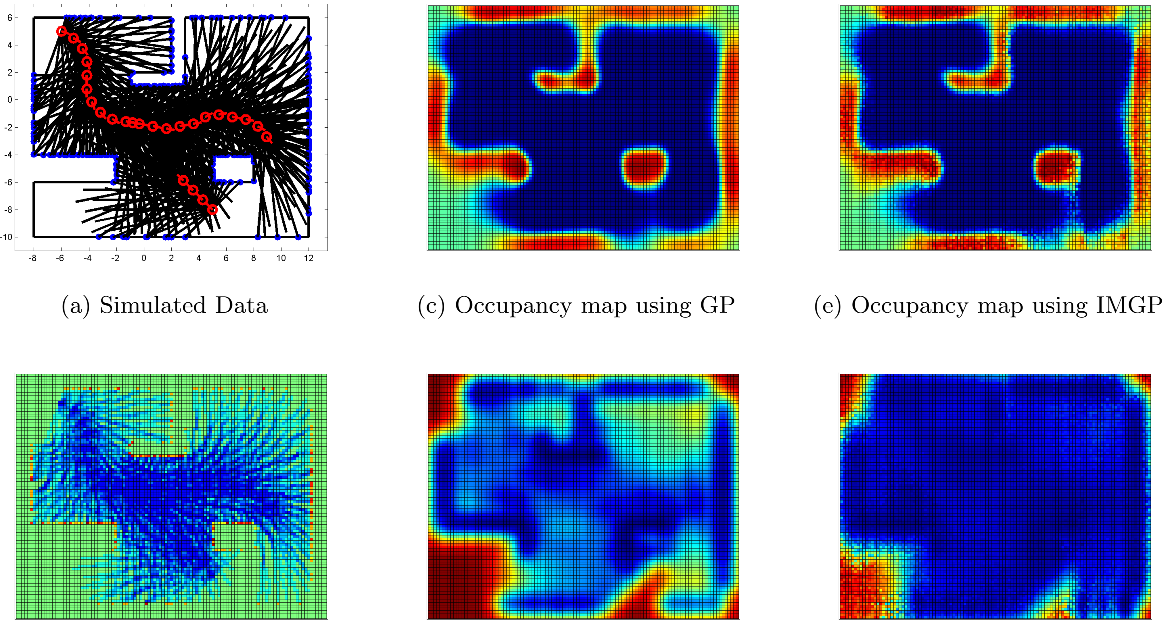
```
 1: Label laser hit points as 1.
 2: Label laser beam segments as $-$length.
 3: Initially partition training data if needed.
 4: Train individual Gaussian process experts $\theta$.
 5: **repeat**
 6:     Gibbs sample training indicators $\mathbf{c}$ with Eq. 17.
 7:     Train individual Gaussian process experts $\theta$.
 8:     Gibbs sample test indicators $\mathbf{c}^*$ with Eq. 16.
 9:     Predict a occupancy map with Eq. 8.
10:     Take the median of the occupancy maps so far.
11: **until** the occupancy map converges.
```

4 Experimental Results

4.1 Toy Example

Fig. 2 compares the regression accuracies of a Gaussian process and an infinite mixture of Gaussian processes with a one dimensional example. We define the ground truth with three distinct functions, $y = 0.1\sin(4\pi x)$ (red), $y = 40x^4 - 141x^3 + 131x^2 - 30$ (green) and $y = 2\sin(8\pi x - 16) + 7$ (blue) in three input domains $x \in [0, 1)$, $[1, 2)$ and $[2, 3)$ to test how the models behave with data from different length-scales. 50 points are randomly selected in each region and associated with Gaussian noises of $\sigma_n^2 = 1$, 1.5 and 1, respectively as shown in Fig. 2a.

Fig. 2b plots the regression results of a Gaussian process. The shaded region denotes the twice standard deviation at each input value x. Compared with the ground truth (green), it is shown that a Gaussian process (red) cannot handle diverse length-scales properly, particularly the third region. In contrast, an infinite mixture of Gaussian process experts after 50 iterations with $\alpha = 1, \phi = 0.01$ (red) in Fig. 2c fits well in three regions. Sample predictions are expressed as dots.



(a) Simulated Data

(c) Occupancy map using GP

(e) Occupancy map using IMGP

(b) Occupancy grid map

(d) Map uncertainty using GP

(f) Map uncertainty using IMGP

Figure 3: 2D Simulation Results. (a) Two robots (red circles) obtain laser beams (black lines) and hit points (blue points), (b) The occupancy grid map, (c) and (d) The occupancy map and its uncertainty of the previous approach, (e) and (f) Those of our approach. Note that red/blue areas are occupied/empty in occupancy maps, while red/blue areas are high/low uncertainties in map uncertainties. The map resolution is set to $20m$.

4.2 Simulated Data

Fig. 3a shows our simulated data. The size of the environment is about 22m × 18m. Two robots acquire laser scans in 26 poses. The laser range finder sweeps 180 degrees with 17 beams whose maximum range is 8m. Totally, 254 laser beams with return and 188 laser beams with no-return are obtained.

Comparison of Accuracy

The conventional occupancy grid map shown in Fig. 3b is accurate but sparse. This is because only those cells which the laser beams pass through are updated based on the independency assumption between cells.

The occupancy map using a Gaussian process in Fig. 3c is quite accurate and dense. The detailed structure of the environment such as the L-shaped wall and the box obstacle is well identified. Its map uncertainty shown in Fig. 3d displays high uncertainty where the robot didn't explore.

Our method after 30 iterations shown in Fig. 3e and 3f are as accurate as the previous approach. For the exact comparison of map accuracy we plot the Receiver Operating Characteristic curve in Fig. 4. It is shown that our method(OM_IMGP, red dotted line) is comparable with the previous approach(OM_GP, blue line) and outperforms the occupancy grid map(OGM, black line).

Comparison of Scalability

The computational times of each map building method are described in Table 1. The software program was implemented in MATLAB and executed on a computer with a Intel Core 2 Duo 3.0 GHz CPU and 3.25 GB RAM.

The occupancy grid map(OGM) which requires no learning steps to predict the results is fastest. With partitioned data, our method (OM_IMGP) dramatically reduced the computational time compared with the previous approach (OM_GP).

In case that the training data is equally divided with K clusters, the computational complexity of a Gaussian process is reduced to $O(K(n/K)^3) = O(n^3/K^2)$. Therefore, with larger training data (more clusters) we get more speed-up. Note that our method has a overhead to sample indicator variables and the computational time per iteration is used to measure the performance. This is because the overhead sampling time and the number of iterations have less influences on the performance with large size of data.

5 Conclusions

In this paper, we have proposed a novel occupancy map building method using an infinite mixture of Gaussian

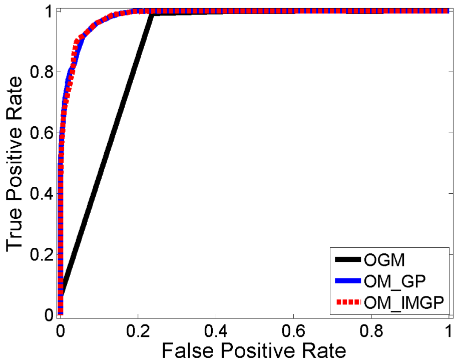


Figure 4: ROC curve

Table 1: Average computational times per iteration of each map building approaches in seconds.

	Clustering	Learning	Prediction	Total
OGM	–	–	0.087 s	0.087 s
OM_GP	–	158.5 s	3.0 s	161.5 s
OM_IMGP	24.8 s	21.5 s	0.4 s	46.7 s

process experts to acquire both accuracy and scalability. Scalability is achieved by clustering training data into subsets with a mixture of experts. Accuracy, on the other hand, is enhanced by considering all the possible clustering configurations with a Dirichlet process.

Simulation results showed that the accuracy of our method is comparable with that of the previous approach, occupancy map building using Gaussian processes, but the computational time of our method is dramatically reduced. Future work includes replacing Gibbs sampling which requires time to converge with approximation methods such as *variational inference* [3] and demonstrating our method with real data.

References

[1] W.M. Gentleman. Implementing Clenshaw-Curtis quadrature, I methodology and experience. *Communications of the ACM*, 15(5):337–342, 1972.

[2] R. Hadsell, J.A. Bagnell, D. Huber, and M. Hebert. Space-carving kernels for accurate rough terrain estimation. *The International Journal of Robotics Research*, 29(8):981–996, 2010.

[3] M.I. Jordan, Z. Ghahramani, T.S. Jaakkola, and L.K. Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

[4] S. Kim and J. Kim. Towards large-scale occupancy map building using Dirichlet and Gaussian processes. In *Proceedings of Australasian Conference on Robotics and Automation*, 2011.

[5] S. Kim and J. Kim. Building occupancy maps with a mixture of Gaussian processes. In *IEEE International Conference on Robotics and Automation*, pages 4756–4761, 2012.

[6] T. Lang, C. Plagemann, and W. Burgard. Adaptive non-stationary kernel regression for terrain modeling. In *Proceedings of Robotics: Science and Systems*, 2007.

[7] H. Moravec and A. Elfes. High resolution maps from wide angle sonar. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 2, pages 116–121, 1985.

[8] R.M. Neal. Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, pages 249–265, 2000.

[9] S. O'Callaghan and F.T. Ramos. Continuous occupancy mapping with integral kernels. In *Proceedings of AAAI Conference on Artificial Intelligence*, pages 1494–1500, 2011.

[10] S. O'Callaghan and F.T. Ramos. Gaussian process occupancy maps. *The International Journal of Robotics Research*, 31(1):42–62, 2012.

[11] S. O'Callaghan, F.T. Ramos, and H. Durrant-Whyte. Contextual occupancy maps using Gaussian processes. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1054–1060, 2009.

[12] J.C. Platt. Probabilities for SV Machines. *Advances in Large Margin Classifiers*, pages 61–74, 2000.

[13] C.E. Rasmussen and Z. Ghahramani. Infinite mixtures of Gaussian process experts. *Advances in Neural Information Processing Systems*, 2:881–888, 2002.

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

[15] S. Thrun. Learning occupancy grid maps with forward sensor models. *Autonomous Robots*, 15(2):111–127, 2003.

[16] V. Tresp. Mixtures of Gaussian processes. *Advances in Neural Information Processing Systems*, pages 654–660, 2001.

[17] K.M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard. Octomap: A probabilistic, flexible, and compact 3D map representation for robotic systems. In *Proceedings of the ICRA workshop on best practice in 3D perception and modeling for mobile manipulation*, 2010.