

# Gaussian process occupancy maps for dynamic environments

Simon T. O’Callaghan<sup>1\*</sup> and Fabio T. Ramos<sup>2</sup>

<sup>1</sup> NICTA

<sup>2</sup> Australian Centre for Field Robotics, School of Information Technologies,  
The University of Sydney, Australia  
[simon.ocallaghan@nicta.com.au](mailto:simon.ocallaghan@nicta.com.au), [fabio.ramos@sydney.edu.au](mailto:fabio.ramos@sydney.edu.au)

**Abstract.** We present a continuous Bayesian occupancy representation for dynamic environments. The method builds on Gaussian processes classifiers and addresses the main limitations of occupancy grids such as the need to discretise the space, strong assumptions of independence between cells, and difficulty to represent occupancy in dynamic environments. We develop a novel covariance function (or kernel) to capture space and time statistical dependencies given a motion map of the environment. This enables the model to perform predictions on how the occupancy state of the environment will be in the future given past observations. We show results on a simulated environment with multiple dynamic objects and on a busy urban intersection.

## 1 Introduction

The Gaussian process occupancy map (GPOM) [12] is a continuous occupancy representation of the environment that overcome some of the limitations with occupancy grids [4]. The method places a Gaussian process (GP) [15] prior over functions mapping the 2D or 3D space into the probability of occupancy. Both laser beams and laser returns are used as free-space and occupied observations respectively to train a GP classifier. The resulting model is not limited to a particular resolution and naturally captures spatial relationships between data points, offering a principled methodology to reason about occlusions, and informative exploration strategies [7], [1].

Despite overcoming some of the issues present in discrete representations such as occupancy grids, GPOM assumed a static world and ignored the effects of time on the model. In reality, mobile robots frequently operate in dynamic environments where the motion of pedestrians, vehicles and other moving objects play an important role in affecting the state of the world and consequently in determining the outcomes of the automaton’s actions. In this work, we extend the GPOM to incorporate a temporal dimension and predict the probability of occupancy at a point in time and space using information from past observations.

Commonly, dynamic entities are represented as a single point (generally the estimated centroid of the object [16]) and observations associated with moving objects are removed before the remaining laser hits

---

\* Formally affiliated with the Australian Centre for Field Robotics, The University of Sydney, Australia.

are passed to a static-world mapper. Recently, a number of works have focused on an alternative approach to classical object tracking by modelling motion on a “sub-object” level within geometric maps. The work in [3] and [11] augment the properties of the occupancy grid map so as to model predicted velocity in a cell as well as its occupancy. In this model, concepts such as objects or tracks do not exist. They are replaced by properties such as occupancy or risk, which are directly estimated for each cell of the grid using sensor observations from consecutive scans to propagate motion and hence the cells’ probabilities of occupancy into the future. Gindele et al. [8] add prior information of the scene to this model in order to predict more complex mobile object motions such as vehicles turning at corners, etc.

Here, we adopt a similar approach by modelling motion on a sub-object level while conserving the continuous nature of our technique and the associated benefits of operating within a fully Bayesian setting. We adapt the framework of a Gaussian process classifier to account for the effects of motion and, consequently, to learn dependencies between consecutive observations to model occupancy in dynamic environments. Our proposed approach learns the dynamic regions of the map and expands on the GPOM, enabling it to propagate the hypothesis of occupancy temporally as well as spatially within a single Bayesian model.

## 2 Dynamic Gaussian Process Occupancy Maps

In GPOM we model the probability of occupancy at a query location  $\mathbf{x}_*$  given a set of  $N$  observations  $\{\mathbf{x}_i, y_i\}_{i=1}^N$ , as  $p(y|X, \mathbf{y}, \mathbf{x}_*) = \Phi(\frac{\alpha\mu_* + \beta}{1 + \alpha^2\sigma_*^2})$ , where  $\Phi(\cdot)$  is the cumulative Gaussian distribution,  $\mu_*$  and  $\sigma_*^2$  are the predictive mean and variance for the posterior of  $\mathbf{x}_*$  respectively, and  $\alpha, \beta$  are parameters of the cumulative Gaussian estimated through leave-one-out cross validation [12]. The posterior is obtained from a  $\mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$ , with mean function  $m(\mathbf{x})$  and covariance function  $k(\mathbf{x}, \mathbf{x}')$ . In this work we generalise GPOM to handle dynamic environments by jointly learning occupancy properties with a motion model of the environment. A block diagram illustrating the proposed inference framework is presented in Figure 1.

### 2.1 Motion-model adapted Gaussian process

Defining  $\Upsilon(\cdot)$  as a vector field estimate of the mean velocity or drift of an underlying function, we can compute the spatial displacement between instants  $t_{\mathbf{x}}$  and  $t'_{\mathbf{x}}$  of a particle originally at location  $\mathbf{x}$  as,

$$\psi(t_{\mathbf{x}}, t'_{\mathbf{x}}) = \int_{t_{\mathbf{x}}}^{t'_{\mathbf{x}'}} \mathbf{v}(t) dt, \quad (1)$$

where  $\mathbf{v}(t)$  is obtained from the velocity vector field. We can then modify the covariance function to take into account displacements. For example, for the squared exponential covariance function, the dynamic version takes the form,

$$k(\hat{\mathbf{x}}, \hat{\mathbf{x}}') = \sigma_f^2 \exp\left(-\frac{|\hat{\mathbf{x}} - A\hat{\mathbf{x}}'|^2}{2L^2}\right), \quad (2)$$

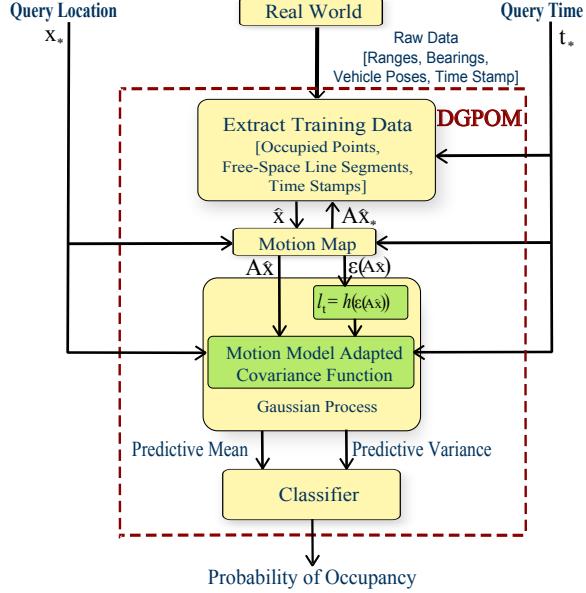


Fig. 1: Block diagram of the inference procedure for a GPOM with motion-map adapted covariance function.

where  $L = \begin{bmatrix} L_s & 0 \\ 0 & l_t \end{bmatrix}$ ,  $A = \begin{bmatrix} 1 & 0 & \psi(t_x, t_{x'})_1 \\ 0 & 1 & \psi(t_x, t_{x'})_2 \\ 0 & 0 & 1 \end{bmatrix}$ ,

and  $\hat{x}$  is the concatenation of the observation's location input vector,  $x$  and  $t_x$ .  $L_s$  is length-scale hyperparameter matrix pertaining to the spatial dimensions and  $l_t$  is the temporal length-scale hyperparameter.

Incorporating a motion model into the covariance function enables the GP to learn dependencies in the observations along the direction of motion rather than along the temporal axis as can be seen in Fig. 2a and 2c. During the training phase, both the spatial and dynamic elements of the model learnt jointly. Improving the estimate of the underlying function's motion, increases the marginal likelihood of the Gaussian process. Similarly, optimising the GP's spatial representation of the function allows for a better alignment of consecutive sets of observations using the motion model.

Motion-model adapted covariance functions offer some useful capabilities in terms of training a Gaussian process to represent the spatial and dynamic behaviour of an underlying function. However, using such a GP to model occupancy in a map proves problematic due to the model's assumption that the entire function is subject to the same motion.

A possible solution to the limitations of the technique is to modify the displacement function so that it is also dependent on location as well as

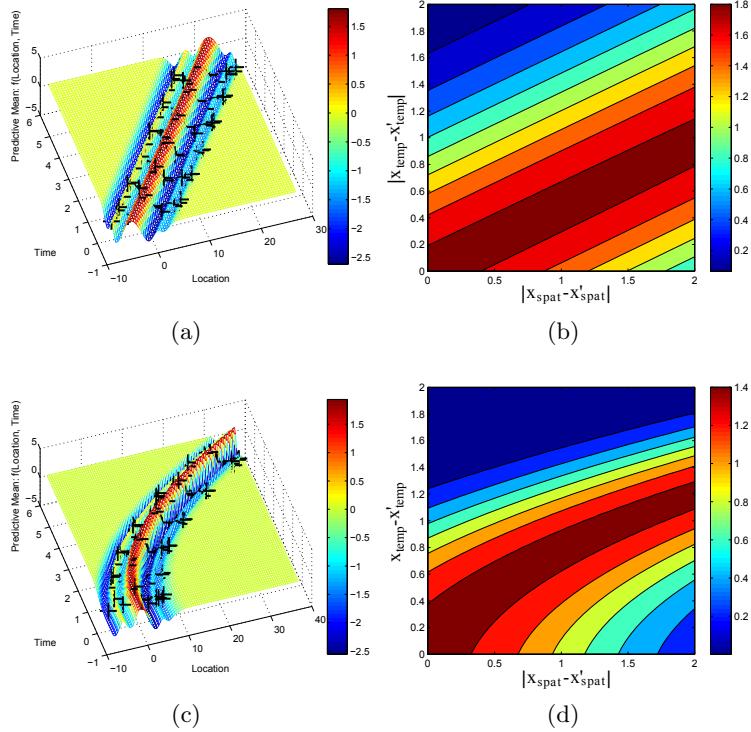


Fig. 2: Predictive mean functions (left column) and their associated covariance contours (right column) after training. The results shown are from a GP employing a constant velocity motion model (top row) and a constant acceleration model (bottom row) for  $\psi(t_{\mathbf{x}}, t_{\mathbf{x}'})$  in Eqn. 2. Observations of the function are represented as black crosses.

time,  $\psi(t_{\mathbf{x}}, t_{\mathbf{x}'}, \mathbf{x})$ . The notion of associating a location with information about the motion of objects at that point is not a new one. Gindele et al. [8] and Brechtel et al. [2] predict the hypothesis of occupancy based on past map states and on velocity values assigned to each cell using prior map knowledge such as a GPS nav-map to propagate occupancy in dynamic regions of the environment. Ellis et al. [5] and O'Callaghan et al. [13] use information from pedestrian trajectory traces to learn motion maps for the environment with applications in object tracking, anomaly detection and path planning. In our case, we seek to learn a motion map that will enable the GP to shape its covariance contours and to determine the extent of temporal dependencies.

## 2.2 Motion-map adapted Gaussian process

Remodelling Eqn. 2 to accommodate for spatial changes in velocities, for the 2-D case, we obtain:

$$A = \begin{bmatrix} 1 & 0 & \psi(t_{\mathbf{x}}, t_{\mathbf{x}'}, \mathbf{x})_1 \\ 0 & 1 & \psi(t_{\mathbf{x}}, t_{\mathbf{x}'}, \mathbf{x})_2 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3)$$

As before, the input vector,  $\mathbf{x}'$ , is translated using  $A$  when used as training data in the Gaussian process. The displacement vector  $\psi(t_{\mathbf{x}}, t_{\mathbf{x}'}, \mathbf{x})$  represents the spatial displacement undergone by a point initialised at  $\mathbf{x}$  over a time interval of  $t_{\mathbf{x}'} - t_{\mathbf{x}}$ . Crucially, its value now also depends on the initial location of the observation rather than simply the time stamps,

$$\psi(t_{\mathbf{x}}, t_{\mathbf{x}'}, \mathbf{x}) = \int_{t_{\mathbf{x}}}^{t_{\mathbf{x}'}} \mathbf{v}(\mathbf{x}(t)) dt - \mathbf{x}, \quad (4)$$

where the velocity,  $\mathbf{v}(\mathbf{x})$ , at any point is governed by the motion map. The estimated motion map will inevitably have some degree of error in it and so it is important that this uncertainty is reflected in the GPOM's hypothesis of occupancy estimates. Consequently, we employ a non-stationary covariance function with the ability to locally alter its temporal length-scale,  $l_t(\mathbf{x})$ , based on the observed quality of the motion map in each region.

[10] provide a comprehensive list of analytical solutions for many popular stationary covariance function's non-stationary form. Here, we use a summation of two non-stationary Matérn class covariance functions with  $\varsigma$  values of 3/2 (Eqn. 5) to model the covariance in the temporal domain as it provides a good balance between capturing sudden changes in the function while also learning long-term trends of the data.

$$k(x, x'; l_t(x), l_t(x')) = \sigma^2 \left( l_t(x) \exp\left(-\sqrt{3} \frac{|x - x'|}{l_t(x)}\right) - l_t(x') \exp\left(-\sqrt{3} \frac{|x - x'|}{l_t(x')}\right) \right), \quad (5)$$

where  $\sigma^2 = 2\sqrt{l_t(x)l_t(x')}/(l_t(x)^2 - l_t(x')^2)$ .

## 2.3 Learning the Motion Map

Motion-map adapted Gaussian processes enable the GPOM to handle dynamic objects by propagating the effects of the movement into the inference model. In this section, we discuss one possible method for deriving the motion map which we represent here as a mean velocity field  $\mathcal{T}(\cdot)$  and an associated error field  $\varepsilon(\cdot)$  based on previous observations of the environment.

The procedure initially builds an occupancy map for each scan. A large body of literature exists on various optical flow techniques, [6] and [9], for extracting regions of motion between two images. Here we opt for a

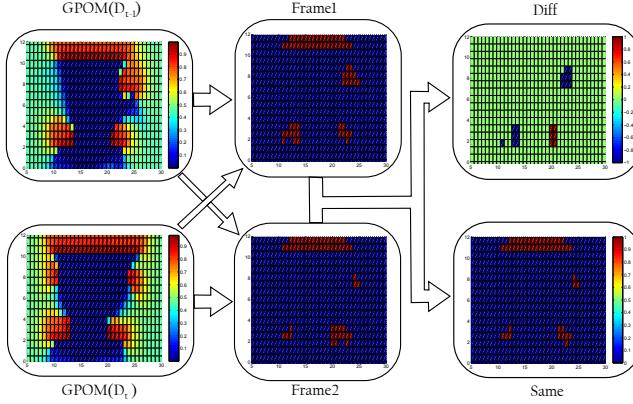


Fig. 3: The stages involved in generating the motion clusters. First stage: Occupancy maps for 2 consecutive scans. Second stage: *Frame1* and *Frame2* of the respective scans. Third stage: *Diff* and *Same* maps.

straightforward and fast agglomerative clustering of query points deemed occupied followed by data association to estimate changes between scans.

Comparing two consecutive maps the algorithm identifies query points in each that have a high probability of being occupied and also have a high probability of being either occupied or free space in the other map. The resulting maps, referred to as *Frame1* and *Frame2* in the figure, are then subtracted from each other and element-wise multiplied by one another to produce *Diff* and *Same*, respectively. The *Diff* map contains clusters of value  $-1$  representing an area that an object has just vacated and clusters of value  $+1$  in an area where an object has just moved into. *Same* highlights regions that have remained occupied in both scans and may possibly contain a stationary object. Sample outputs of these 3 stages are shown in Fig. 3.

The algorithm then clusters the positive valued cells as objects and the non-zero cells from *Diff* and *Same* as motion clusters. For each motion cluster in *Diff*, the algorithm searches for an adjacent object. If one is found, this object is assumed to have either just vacated or occupied that cluster depending on the value of that cluster ( $+1$  or  $-1$ ). The direction of motion is obtained by calculating the angle between the centroid of the motion cluster and the associated object cluster. The magnitude of the motion is simply the width of the cluster. Stationary objects are identified as clusters in *Same* that do not have a neighbouring motion cluster from *Diff*.

These clusters and their velocities are then stored in memory and the procedure is repeated for each pair of consecutive scans in the training set. A regressor is trained using a portion of these clusters to infer a velocity at any point in space, i.e. the mean velocity field  $\Upsilon(\cdot)$ . Finally, the error of this field is assessed using clusters withheld from the training and a second regressor is then trained to approximate our estimated accuracy of  $\Upsilon(\cdot)$  over the entire region.

## 2.4 Relating $\varepsilon(\cdot)$ to $l_t(\cdot)$

Theoretically, the latent temporal length-scale hyperparameter function,  $l_{t(x)}$ , can be learnt while maximising its marginal likelihood such as in [14]. Implementing such a scheme unfortunately is quite expensive due to the large search space for learning the parameters of the function combined with the relatively slow iteration time of the optimiser due to the requirement of inverting an  $n \times n$  covariance matrix each time.

A convenient approximation is to learn a parametric mapping,  $h(\cdot)$ , from a point's estimated translation error,  $\varepsilon(x)$  to the temporal length-scale,

$$l_t(x) \approx h(\varepsilon(x)) = a * \varepsilon(x)^{-(\frac{1}{b})}. \quad (6)$$

The chosen form of the mapping is described by Eqn. 6. The parameter  $b$  controls the rate of decay of  $l_t$  as  $\varepsilon(x)$  increases while  $a$  serves as a scaling parameter.

This approach to learning the temporal length-scale results in two additional dimensions being added to the search space during training. However, it is a significantly more constrained problem than attempting to train a latent non-parametric function,  $l_t(\cdot)$  during optimisation of the marginal likelihood.

## 3 Experiments and Results

In this section, we examine the performance of the proposed DGPM algorithm. Initially, we use a synthetic dataset simulating the observations received from a car-mounted laser rangefinder sensor positioned at a T-junction. The second experiment involved gathering range data at a city intersection. Both experiments include quantitative and qualitative analysis.

### 3.1 Simulated Data Experiment

Figure 4 presents a summary of a synthetic scenario. The rangefinder sensor is positioned at  $x = (15; 0)$  and observes cars passing the junction at velocities of 2 m/s or -1 m/s depending on which lane the vehicle occupies. Each scan contains 70 beams covering a 180° sweep with a maximum range of 20 metres sampled at a frequency of 1 Hz.

**The motion map**  $\Upsilon(\cdot)$  and  $\varepsilon(\cdot)$  for the environment are learnt with data acquired a priori. The motion clusters and their velocities are shown in Fig. 5a. A subset consisting of 500 of these points was used to train a regressor to model the horizontal and vertical components of  $\Upsilon(\cdot)$ . Fig. 5b presents the regressor's output for the horizontal component of the estimated mean velocity field. Both lanes are clearly distinguishable from the plot with the regressor also learning that vehicles observed in the lane traveling from left to right tended to move at approximately twice the speed of the vehicles in the opposite lane. The mean velocity field also estimates an average velocity of 0 m/s in the area occupied by the building at the top of the scene. A quiver plot of the resulting mean velocity field is superimposed onto the environment in Fig. 5c.

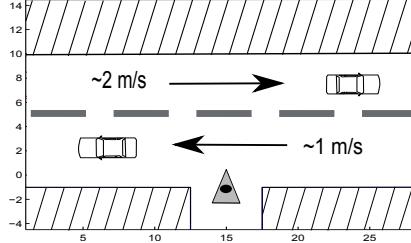
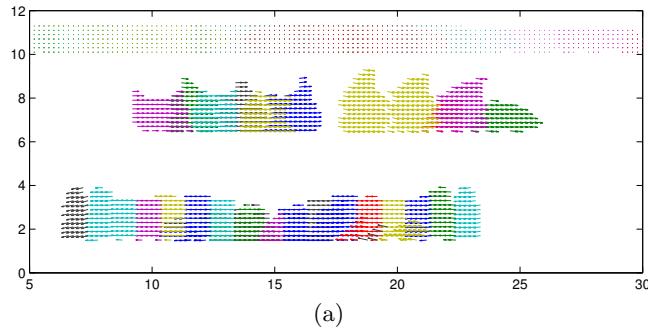
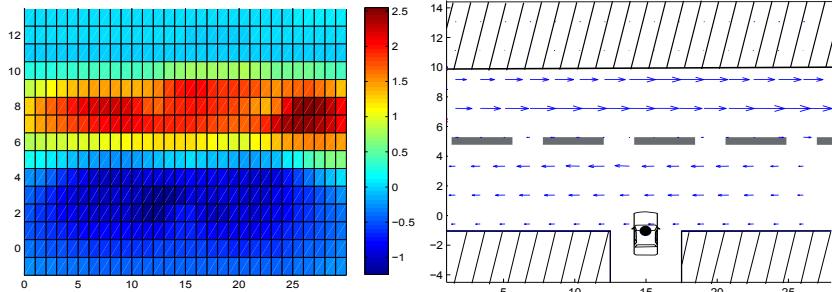


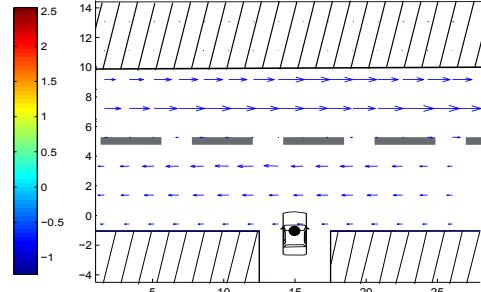
Fig. 4: Ground truth of simulated experiment.



(a)



(b)



(c)

Fig. 5: a) The resulting motion clusters with their associated velocity vectors (dynamic and static). b) Output of regressor modelling the horizontal component of  $T(\cdot)$ . c) A quiver plot of the motion map's velocity superimposed on the environment.

**DGPOM outputs** Using the Dynamic GPOM framework the probability of occupancy can be inferred at various instances in time and space. The outputs from two of those instances are displayed in Fig. 6. The ground truths at  $t = 3$  and  $t = 5$  (top row) show two vehicles in each lane (blue rectangles) with a large amount of occlusion created by the cars closer to the sensor. The red lines represent the range-finder sensor's observations at that time step. Incorporating observations from previous

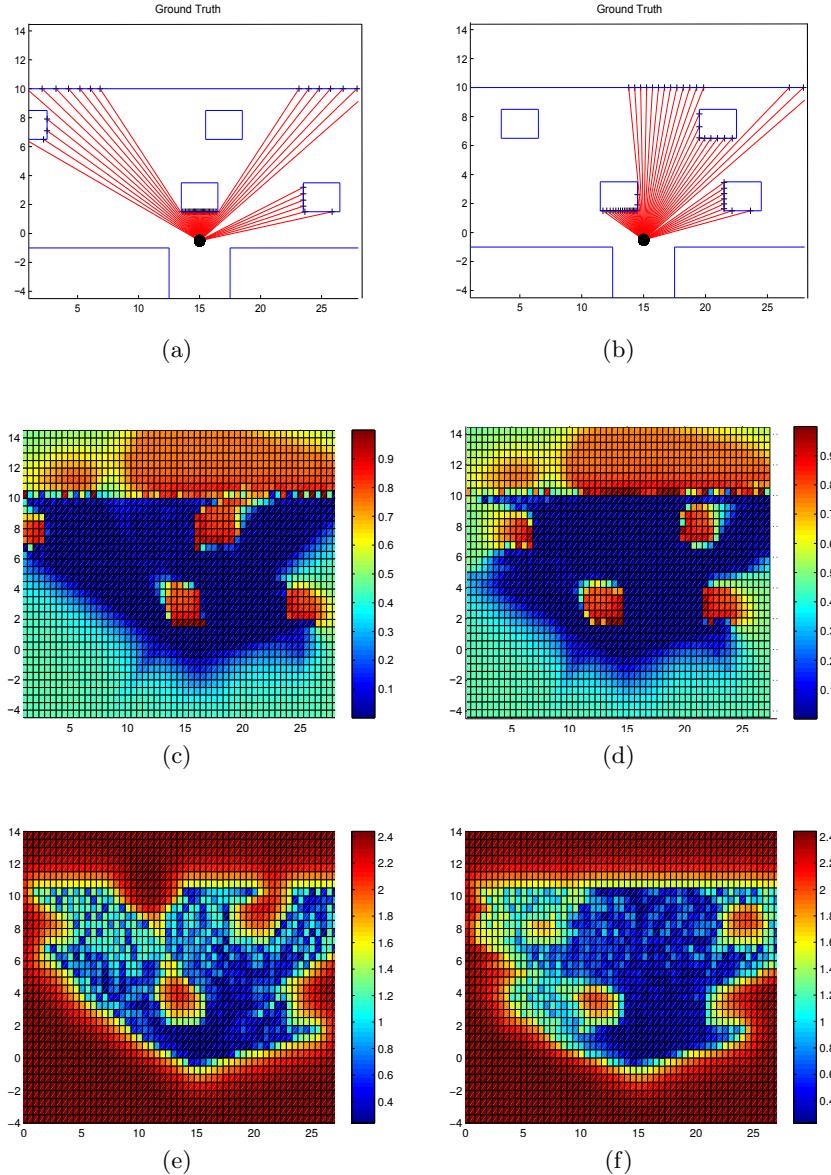


Fig. 6: Results for simulated experiment at two separate instances in time.  $t = 3$  (left column) and  $t = 5$  right column. Top: Ground truth of the environment's state. Middle: Probability of occupancy versus location which incorporates observations from previous time steps. Bottom: Predictive variance maps.

scans using the DGPOOM framework enables the algorithm to accurately infer a large amount of the scene (middle row). Despite not observing the car at (17; 7.5) in the first image directly, the procedure uses the learnt motion map to infer a strong covariance between that location and previous observations of the vehicle resulting in a high probability of occupancy in the mapper’s output at the aforementioned location. Similar behaviour can also be seen in the second column at (5; 7.5). The associated predictive variance maps (bottom row) offer an insight into how the scene was reconstructed. The dark blue regions of high confidence rely primarily on the most recent observations. Light blue and green areas indicate estimates in regions that have not been observed in the most recent scans and, due to a degree of inaccuracy in motion map,  $\gamma(\cdot)$ , are less confident in the hypothesis of occupancy.

The additional flexibility afforded to the algorithm by incorporating location as an extra parameter in determining the behaviour of the covariance function enables the GP to exploit strong temporal dependencies between scans containing multiple dynamic and stationary objects. As a result, the probability of occupancy for the entire region can be handled within the same Bayesian setting without the need for filtering out dynamic objects and handling them in a separate procedure. Large covariances along traffic lanes learnt by the covariance function allows the GP to accurately infer the location of the car by propagating the influence of past observations forward through time using the velocity vectors of the motion map.

Table 1: Comparison of areas under the ROC curve at various instances in time (columns) using 3 different approaches.

<b>Algorithm</b>	<b>Query Time (<math>(t_{x_*})</math>)</b>						
	$t_{x_*} = 3$	$t_{x_*} = 4$	$t_{x_*} = 5$	$t_{x_*} = 6$	$t_{x_*} = 7$	$t_{x_*} = 8$	$t_{x_*} = 9$
DGPOM $\Re^{D+1}$	0.9497	0.9301	0.9300	0.9288	0.9480	0.9372	0.9435
GPOM $\Re^D$	0.7736	0.8040	0.8248	0.8227	0.8364	0.8376	0.8850
GPOM $\Re^{D+1}$	0.8383	0.8773	0.8903	0.8954	0.9267	0.8922	0.8974

**ROC Tests** With a known ground truth it is possible to determine the precision of the estimates using the ROC curves once more. Table 1 shows the results of a comparison between the inference algorithm’s outputs and two variations of the standard GPOM; GPOM ( $\Re^D$ ) employs a D-dimensional Gaussian process and hence ignores the observations’ time stamps while GPOM ( $\Re^{D+1}$ ) includes time as an additional feature in the classic GP architecture. The table lists the areas under the ROC curve produced by each algorithm over a series of time steps with the DGPOM consistently outperforming its static counterparts. All three algorithms utilise rangefinder data acquired at the query time ( $t_{x_*}$ ) and three previous scans. While the hypothesis of occupancy for the GPOM ( $\Re^D$ ) in motionless regions such as the building wall is comparable to the DGPOOM, the static-world assumptions it makes result in an inability to

reconstruct the vehicles accurately. The dynamic objects in the scene also negatively influence the performance of the GPOM ( $\Re^{D+1}$ ) leading to a short temporal length-scale. Consequently, only the observations acquired at the query instances have any significant influence on the probability of occupancy estimate.

### 3.2 Real Data Experiment

A Pioneer 2-AT robot equipped with a SICK LMS291 laser rangefinder gathered observations from 3 minutes of traffic flow at a busy intersection. Fig. 7 provides an aerial view of the area including the location of the robot during testing. Superimposed on the image are the laser returns from the static objects (manually classified) and the mean velocity field. The road lanes can be identified clearly in the plot as well as a number of zero-magnitude velocity vectors around the buildings.

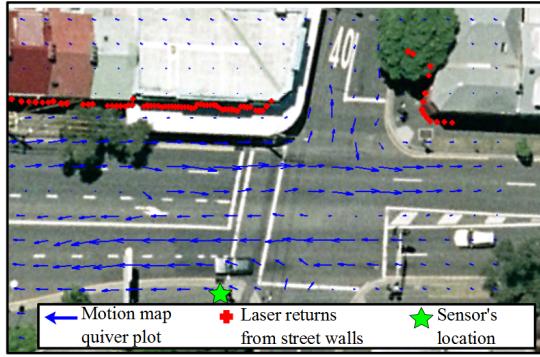


Fig. 7: Aerial image of the outdoor environment used in the experiment with the learned velocity field superimposed on top of it.

A contour plot generated from sampling the  $\varepsilon(\cdot)$  function has also been superimposed on the aerial image in Fig. 8. Comparing the  $\Upsilon(\cdot)$  with motion clusters from a test set indicated a degree of error in the mean velocity field along the roadways as well as a large error in the region of the intersection. A certain level of error is to be expected due to variations in speed between vehicles however the considerable error over the junction is primarily due to the fact that the direction of motion in this region can fluctuate greatly here. Consequently, the local temporal length-scale in this region will be affected, resulting in past observations having a comparatively small influence on the hypothesis of occupancy. The parameters of Eqn. 6 converged to  $a = 0.8625$  and  $b = 1.242$  after training corresponding to an  $l_t < 0.5$  across the centre of the intersection. A possible solution to address is discussed in Section 4.

The outputs of the DGPM's inference algorithm at four time steps and the observations acquired at each instant are displayed in Fig. 9. To illustrate the sense of motion in the estimates, the outputs in the second and

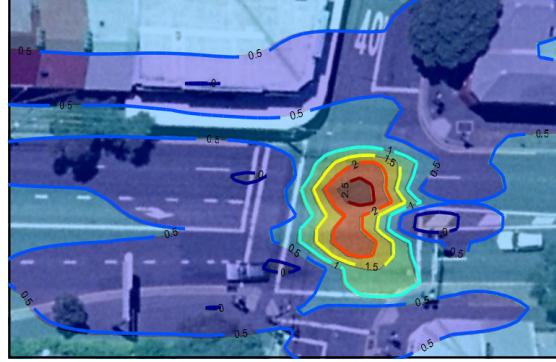


Fig. 8: Contours from the error map superimposed over the environment.

fourth row are at a query time of one time step after the first and third rows, respectively. Similar to its performance on the synthetic dataset, the algorithm's probability of occupancy map predicts high probabilities in locations occupied by vehicles despite being occluded from the sensor at the  $t_{x_*}$ . In the first set of outputs, the algorithm predicts the presence of two cars at the center of map,  $\mathbf{x}_* \approx (0; 13)$ , using data from previous scans. There is an increase in uncertainty in the second set of prediction as the time since both cars were last observed increases. The accuracy of the  $\Upsilon(\cdot)$  in some areas enables information obtained from scans in the past to accurately predict the location and outline of both buildings on either side of the intersection wall.

To analyse the performance of the DGPOM's predictions as time since the last observation increases, a ground truth is required. Although the exact state of the environment at any given instant is unknown, it is approximated by an occupancy map generated using observations acquired exclusively at the given  $t_{\mathbf{x}_*}$  and comparisons are made only in areas where this map has a high degree of confidence. Fig. 10 shows the averaged behaviour of the area under the ROC curve as time since the last observation is increased. Initially, there is a shape falloff in performance mainly due to the predicted probability of occupancy in regions such as the centre of the intersection rapidly reverting to the global mean of 0.5. Eventually the curve begins to saturate once the dynamic regions return to a global mean due to a lack of new observations while the hypothesis of occupancy areas of the map believed to be static remains confident. The y-axis is scaled from  $0.5 \rightarrow 1$  to represent the range from a random guess to a perfect reconstruction of the approximated ground truth.

## 4 Conclusions

In this paper, we introduced a version of the GPOM algorithm to deal with dynamic environments. We developed a continuous occupancy map capable of learning static and dynamic regions and integrating observations from multiple points in time into a single continuous probabilistic spatio-temporal model of the environment. The proposed motion-model

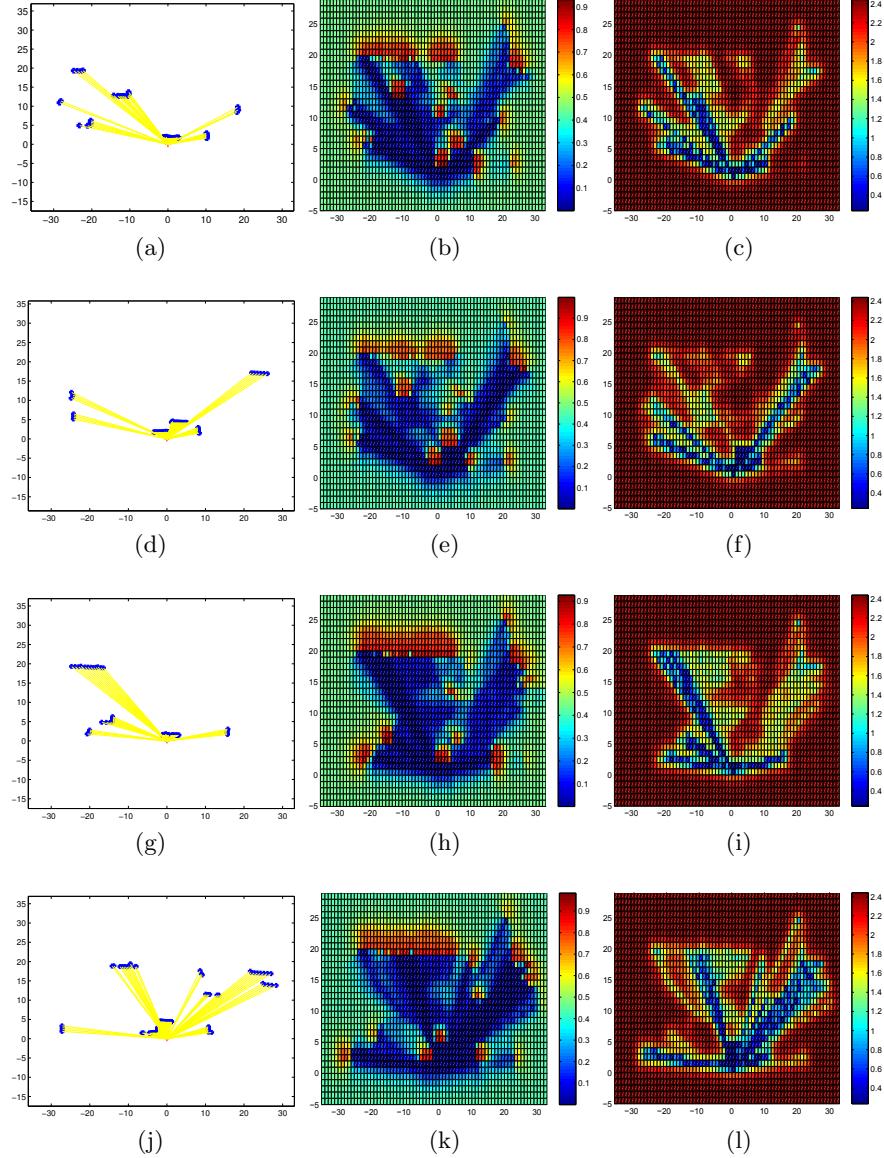


Fig. 9: Sequence of images illustrating the output of the DGPOm at different instances in time. From top to bottom:  $t = 4$ ,  $t = 5$ ,  $t = 29$ ,  $t = 30$ . Left column: Range observations recorded at each time stamp. Middle column: Probability of occupancy versus location. Right column: Predictive variance verus location.

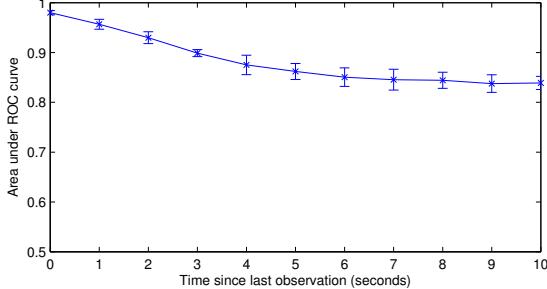


Fig. 10: Averaged decay and saturation of the area under the ROC curve as the time since the last observation increases. The error bars represent the 1 standard deviation boundary.

adapted covariance function enables the parameters of an equation describing the drift of an underlying function to be learnt in tandem with the hyperparameters by maximising the marginal likelihood. The benefits of this framework could make interesting contributions to modelling phenomena that vary in space and time while also being subject to drift such as air pollution.

However, to model spatially non-uniform drift (multiple dynamic objects), the search space for jointly optimising the GP and latent displacement function becomes infeasibly large. Consequently, we constrain the problem by assuming the function or motion map,  $\Psi(\cdot)$ , is temporally intransigent. Dependencies between observations are propagated temporally as well as spatially while employing a non-stationary covariance function to adjust their magnitude depending on the estimated accuracy of the translation  $Ax$ .

Results using the DGPM are encouraging although the implementation will need to exploit its parallelisable properties using multiple cores to be used in an online setting effectively (similar to [3], [8], [2], [11]). Static obstacles such as buildings are represented clearly despite considerable occlusions. In dynamic regions, the motion of cars is inferred using past observations and the motion map. The hypothesis of occupancy is weighted by the estimated error of the mean velocity assumption to prevent inaccurate modelling of the motion leading to the creation of phantom obstacles.

## References

1. J. Bohg, M. Johnson-Roberson, M. Björkman, and D. Kragic. Strategies for multi-modal scene exploration. In *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010.
2. S. Brechtel, T. Gindele, and R. Dillmann. Recursive importance sampling for efficient grid-based occupancy filtering in dynamic environments. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 3932 –3938, May 2010.

3. C. Chen, C. Tay, C. Laugier, and K. Mekhnacha. Dynamic environment modeling with gridmap: A multiple-object tracking application. In *Control, Automation, Robotics and Vision, 2006. ICARCV '06. 9th International Conference on*, pages 1 –6, Dec. 2006.
4. A. Elfes. *Occupancy grids: a probabilistic framework for robot perception and navigation*. PhD thesis, Carnegie Mellon University, 1989.
5. D. Ellis, E. Sommerlade, and I. Reid. Modelling pedestrian trajectories with gaussian processes. In *Ninth International Workshop on Visual Surveillance*, 2009.
6. D. Fleet and Y. Weiss. *Optical flow estimation*. 2005.
7. S. Gan, K. Yang, and S. Sukkarieh. 3d path planning for a rotary wing UAV using a Gaussian process occupancy map. In *Proceedings of the Australasian Conference on Robotics and Automation*, 2009.
8. T. Gindele, S. Brechtel, J. Schroder, and R. Dillmann. Bayesian occupancy grid filter for dynamic environments using prior map knowledge. In *Intelligent Vehicles Symposium, 2009 IEEE*, pages 669 –676, June 2009.
9. B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, pages 674–679, San Francisco, CA, USA, 1981. Morgan Kaufmann Publishers Inc.
10. A. Melkumyan and F. Ramos. Multi-kernel gaussian processes. In *IJCAI*, pages 1408–1413, 2011.
11. J. Moras, V. Cherfaoui, and P. Bonnifait. Credibilist occupancy grids for vehicle perception in dynamic environments. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 84 –89, May 2011.
12. S. T. O’Callaghan and F. T. Ramos. Gaussian process occupancy maps. *The International Journal of Robotics Research*, 31(1):42–62, 2012.
13. S. T. O’Callaghan, S. P. N. Singh, A. Alempijevic, and F. T. Ramos. Learning navigational maps by observing human motion patterns. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 4333 –4340, May 2011.
14. C. Plagemann, K. Kersting, and W. Burgard. Nonstationary Gaussian Process Regression using Point Estimates of Local Smoothness. In *European Conference on Machine Learning (ECML)*, 2008.
15. C. E. Rasmussen. *Gaussian Processes for Machine Learning*. MIT Press, 2006.
16. C. C. Wang. *Simultaneous Localization, Mapping and Moving Object Tracking*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, April 2004.