Dynamic Hilbert Maps:
Real-Time Occupancy Predictions in Changing Environments

Vitor Guizilini[1,2], Ransalu Senanayake[1], and Fabio Ramos[1,3]

***Abstract*— This paper proposes a technique for predicting future occupancy levels. Due to the complexity of most real-world environments, such as urban streets or crowded areas, the efficient and robust incorporation of temporal dependencies into otherwise static occupancy models remains a challenge. We propose a method to capture the spatial uncertainty of moving objects and incorporate this information into a continuous occupancy map represented in a rich high-dimensional feature space. Experiments performed using LIDAR data verified the real-time performance of the algorithm.**

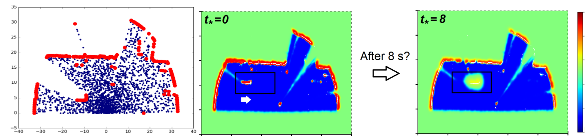
I. INTRODUCTION

Occupancy maps which discern free areas of the environment (safe for traversal) from occupied areas (would result in a collision) are commonly used in autonomous vehicles. Straightforward approaches to static occupancy mapping rely on a grid-based non-overlapping discretization of the environment [1]. Because grid cells are updated individually without considering the relationship among cells, this discretization process completely discards spatiotemporal dependencies. Furthermore, this representation quickly becomes infeasible for larger datasets, especially when dealing with volumetric data. The Hilbert Mapping (HM) framework [2], [3] is an alternative to grid maps and can produce a continuous representation of occupancy states in a much lower computational cost.

Occupancy mapping in dynamic environments can be categorized into three classes: 1) building static occupancy maps in the presence of dynamic objects by considering moving objects as spurious data [4], [5], 2) mapping the long-term dynamics of the environment [6], [7], and 3) mapping the short-term dynamics of the environment. This paper focuses on the third category. Short-term dynamics are important not only for understanding instantaneous changes in the environment, but also using this information to make predictions into the future.

Considering the limitations of previous attempts — dynamic Gaussian processes (DGP) [8] and spatiotemporal Hilbert maps (STHMs) [9] — to predict future occupancy, we propose a novel methodology for spatiotemporal modeling. As illustrated in Fig. 1, the area of future uncertainty around the moving vehicle is much larger due to the inherent unpredictability of future states.

The authors are with the [1]School of Computer Science, at The University of Sydney, Australia; [2]Toyota Research Institute, USA; and [3]NVIDIA Research, USA. Emails: `vitor.guizilini@tri.global`;{ `ransalu.senanayake`;`fabio.ramos` }`@sydney.edu.au`



(a) Dataset at $t = 0$ (b) Future occupancy prediction

Fig. 1: (a) Data-frame captured by a LiDAR (blue: laser beams and red: laser hit points). The vehicle inside the rectangle moves from left to right (b) The current and future (8 seconds) occupancy maps produced by the Dynamic Hilbert maps (DHM) algorithm. Indicating the uncertainty of future predictions, the occupancy probability (red indicates highly probable) of the position of the vehicle and its surrounding is relatively low. The future prediction is represented as a spatial distribution peaked at one point which drops down radially, making such a map ideal for safer path planning.

II. METHOD

Following [10], we define a collection of hinged locations $\tilde{\mathcal{X}}$ that act as centers for anchoring *kernels*. With analogous to a multivariate Gaussian shape, these hinged locations have a center $\boldsymbol{\mu} \in \mathbb{R}^3$ alongside another matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{3\times 3}$ to denote how far the measurements affect in each direction . With the M hinged locations $\tilde{\mathcal{X}} = \{\tilde{\mathbf{x}}_m\}_{m=1}^{M} = \{(\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)\}_{m=1}^{M}$, the occupancy probability of any point in the environment $\mathbf{x}_* \in \mathbb{R}^3$ can be computed using a logistic model,

$$p(y_* = 1|\mathbf{x}_*, \mathbf{w}, \tilde{\mathcal{X}}) = \Big(1 + \exp\Big(-\mathbf{w}^\top \Phi(\mathbf{x}_*)\Big)\Big)^{-1}, \quad (1)$$

with a feature vector defined as:

$$\Phi(\mathbf{x}_*, \tilde{\mathcal{X}}) = [k(\mathbf{x}_*, \tilde{\mathbf{x}}_1), k(\mathbf{x}_*, \tilde{\mathbf{x}}_2), \ldots, k(\mathbf{x}_*, \tilde{\mathbf{x}}_M)], \quad (2)$$

$$k(\mathbf{x}_*, \tilde{\mathbf{x}}_m) = \exp\left(-\frac{1}{2}(\mathbf{x}_* - \boldsymbol{\mu}_m)^\top \boldsymbol{\Sigma}_m^{-1}(\mathbf{x}_* - \boldsymbol{\mu}_m)\right). \quad (3)$$

Refer `https://github.com/RansML/Bayesian_Hilbert_Maps/blob/master/BHM_tutorial.ipynb` for an intuitive explanation. The parameters $\mathbf{w}$ in (1) are learned by maximizing the regularized log-likelihood using stochastic gradient descent (SGD) with hit-free points $\mathbf{x}$ collected from a LIDAR.

Our objective is to build short-term occupancy maps and make short-term predictions into the future. To accomplish this, three different Hilbert maps are maintained: $\mathcal{H}_p$, representing the previous timestep; $\mathcal{H}_c$, representing the current timestep; and $\mathcal{H}_a$, representing the accumulated model that is iteratively constructed as more data is collected. With this, we take the following four steps for each new LIDAR scan.

1. Object segmentation: In order to maintain memory and speed efficiency, we firstly cluster the pointcloud using the Quick-Means algorithm and discard the pointcloud. We obtain P objects $\mathcal{O}_t = \{\mathcal{O}_t^p\}_{p=1}^{P}$.

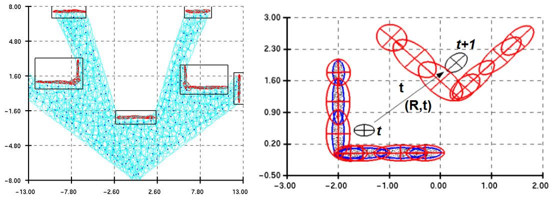


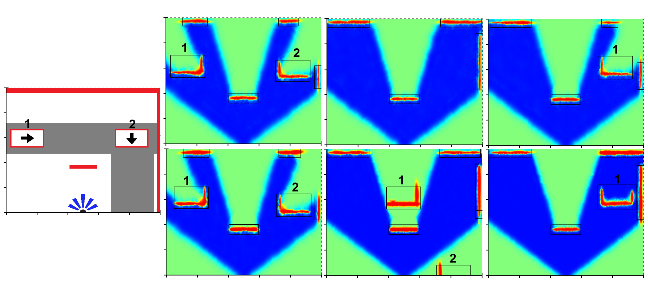
Fig. 2: Example of motion calculation. (a) extracted clusters (including covariance ellipses and object boundaries). (b) Example of how location uncertainty is propagated into mapping. At t the object pointcloud (black dots) is used to generate the cluster set $\tilde{\mathcal{X}}$ (blue ellipses), and the object location uncertainty $\mathbf{P}$ (black ellipse) is propagated to the covariance $\boldsymbol{\Sigma}$ of these clusters (red ellipses). At $t+1$, given the transformation, its location can be estimated and used to determine new cluster positions and covariances.

2. Motion calculation: The motion between $\mathcal{O}_t$ and $\mathcal{O}_{t+1}$ is then computed using the sparse Iterative Closest Point (ICP) algorithm. This results in an orthogonal transformation $\mathbf{R}$ and a translation vector $\mathbf{t}$ in $SO(3)$.

3. Dynamic object model: With $\boldsymbol{\mu}$ defining the location and θ defining the orientation, we define a 6-dimensional state vector $\mathbf{x}_t^p = \{\boldsymbol{\mu}, \theta, \dot{\boldsymbol{\mu}}, \dot{\theta}\}$ for each object $\mathcal{O}_t^p$, alongside its 6×6 covariance matrix $\mathbf{P}_t^p$ (initialized to zero). This state vector, alongside $\mathbf{R}$ and $\mathbf{t}$, is then used for updating a Kalman filter. This dynamic model allows us to: 1) estimate object position in future (or past) timesteps; 2) incorporate new observations to improve predictions; and 3) account for sensor and model uncertainties.

4. Feature vector update: With new updates, $\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t + \mathbf{t}$, $\Sigma_{t+1} = \mathbf{R}_t \Sigma_t \mathbf{R}_t^\top$, and a contribution parameter ω_{t+1} that depends on the cluster covariance, we can now compute the feature vector (2) and learn new weights $\mathbf{w}$ to represent $\mathcal{H}_a$.

Under the DHM framework, it is possible to query the occupancy $p(y_* = 1|\mathbf{x}_*, t_*, \mathbf{w}, \mathcal{H}_p, \mathcal{H}_a, \mathcal{D}_t)$ anywhere in the space $\mathbf{x}_*$ at anytime t_* (past, present, and future). Fig. 3 shows an example of the proposed framework, as introduced in this section.



(a) $t = 0$ (b) $t_* = 0$ (c) $t_* = 18$ (d) $t_* = 42$

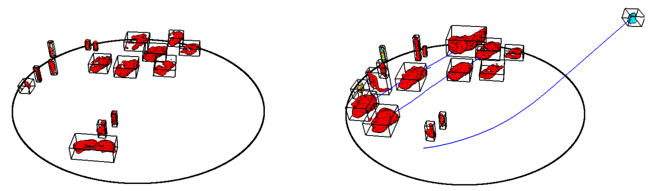
Fig. 3: Example of the proposed DHM framework on simulated 2D data. (a) LiDAR (blue) observing a dynamic environment with two moving vehicles (direction in arrows) and obstacles (red). (b)-(d) Future predictions. The top row shows different $\mathcal{H}_c$, generated from current sensor data, while the bottom row shows the corresponding $\mathcal{H}_a$, generated by incremental propagation between time-steps. (b) indicates $\mathcal{H}_a$ is robust against occlusions, projecting motion into areas outside the field of view. Because of the parameter accumulation process, vehicle 1 in (c) of $\mathcal{H}_a$ has correctly mapped both sides of the vehicle.

III. Experiments

We conducted experiments with toy datasets and the KITTI Vision Benchmark Suite. As shown in Table I and Fig. 4, DHM not only outperforms other methods, but also capable of making future predictions in 3D.

TABLE I: Occupancy prediction comparision (F-Measure)

Time step	Dataset 1 (2D)				Dataset 2 (3D)	
	HM	DGP	STHM	DHM	HM	DHM
$t_* = 0$	0.824	0.788	0.839	0.844	0.843	0.837
$t_* = 1$	0.707	0.752	0.784	0.826	0.727	0.804
$t_* = 3$	0.581	0.678	0.691	0.803	0.529	0.742
$t_* = 5$	0.418	0.571	0.653	0.756	0.472	0.718
$t_* = 10$	0.139	0.419	0.524	0.663	0.189	0.625



(a) $t_* = 0$ s (b) $t_* = 45$ s

Fig. 4: 3D occupancy prediction using DHM. The black circle indicates maximum sensor range (40 m, centered at $(0, 0, 0)$), and blue lines indicate object centroid motion over time. Video and code: `https://bitbucket.org/vguizilini/cvpp`.

Runtime: All computations were performed on a *i7/2.60×8 GHz* notebook, with multi-threading enabled wherever possible. DHM updates between timesteps require roughly 70 ms and 120 ms in 2D and 3D datasets, respectively, which makes it applicable to online tasks under real-time constraints.

References

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

[2] F. Ramos and L. Ott, "Hilbert maps: Scalable continuous occupancy mapping with stochastic gradient descent," in *Proceedings of Robotics: Science and Systems (RSS)*, 2015.

[3] K. Doherty, J. Wang, and B. Englot, "Probabilistic map fusion for fast, incremental occupancy mapping with 3d hilbert maps," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on.* IEEE, 2016, pp. 1011–1018.

[4] D. Meyer-Delius, M. Beinhofer, and W. Burgard, "Occupancy Grid Models for Robot Mapping in Changing Environments," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2012.

[5] D. Nuss, R. Stephan, M. Thom, T. Yuan, K. Gunther, M. Michael, G. Axel, and K. Dietmayer, "A random finite set approach for dynamic occupancy grid maps with real-time application," in *arXiv preprint arXiv:1605.02406*, 2016.

[6] T. Krajnık, P. Fentanes, G. Cielniak, C. Dondrup, and T. Duckett, "Spectral analysis for long-term robotic mapping," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 3706–3711.

[7] R. Senanayake, S. O'Callaghan, and F. Ramos, "Learning highly dynamic environments with stochastic variational inference," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[8] S. O'Callaghan and F. Ramos, "Gaussian process occupancy maps for dynamic environments," in *Experimental Robotics*. Springer Tracts in Advanced Robotics, 2015, vol. 109, pp. 791–805.

[9] R. Senanayake, L. Ott, S. O'Callaghan, and F. Ramos, "Spatio-temporal Hilbert maps for continuous occupancy representation in dynamic environments," in *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[10] V. Guizilini and F. Ramos, "Large-scale 3d scene reconstruction with Hilbert maps," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2016.