

“© 2019 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.”

# Representation of Uncertain Occupancy Maps with High Level Feature Vectors

Janindu Arukgoda<sup>1</sup>, Ravindra Ranasinghe<sup>1</sup> and Gamini Dissanayake<sup>1</sup>

**Abstract**—This paper presents a novel method for representing an uncertain occupancy map using a “feature vector” and an associated covariance matrix. Input required is a point cloud generated using observations from a sensor captured at different locations in the environment. Both the sensor locations and the measurements themselves may have an associated uncertainty. The output is a set of coefficients and their uncertainties of a cubic spline approximation to the distance function of the environment, thereby resulting in a compact parametric representation of the environment geometry. Cubic spline coefficients are computed by solving a non-linear least squares problem that enforces the Eikonal equation over the space in which the environment geometry is defined, and zero boundary condition at each observation in the point cloud. It is argued that a feature based representation of point cloud maps acquired from uncertain locations using noisy sensors has the potential to open up a new direction in robot mapping, localisation and SLAM. Numerical examples are presented to illustrate the proposed technique.

## I. INTRODUCTION

Availability of a map of the environment is crucial for achieving many robotic tasks such as path-planning and localisation. The map of an environment can be represented in many different ways, typically tailored to the task at hand. For example, motion planning requires a map that describes the regions that the robot can freely move avoiding collisions while localisation requires the environment to be represented in a way that the measurements from the sensors on board the robot can easily be related to the map, and navigation in an uneven terrain requires the map to provide a three dimensional description of the environment.

Current simultaneous localisation and mapping (SLAM) algorithms make it possible to acquire maps of environments in a number of ways. When the environment can be represented by the locations of a set of geometric primitives such as points, corners, lines or planes, a feature based SLAM algorithm [1][2] can be used to obtain the locations of such features and their uncertainty. While such feature maps can provide a very compact representation, they do not provide information about the free space that navigation tasks such as motion planning and path planning require. Alternatively, given information captured by a high resolution range sensor such as a laser range finder or a depth camera, it is possible to use a pose-slam algorithm [3] to compute the location of the robot poses from which the sensor observations were acquired. A “point cloud” map of the environment (fig. 1)

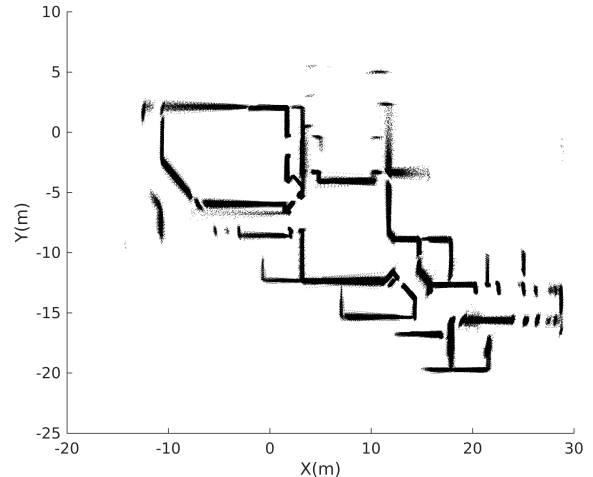


Fig. 1: Point Cloud Map of an Office Space

can then be obtained by transforming all sensor observations into a global coordinate frame. Although the point cloud inherently captures the observation uncertainty, any errors in the robot location from where the data was collected is typically ignored. By converting the point cloud into an occupancy grid map (OGM), a representation suitable for tasks such as localisation, path planning and obstacle avoidance can be obtained. Alternatively Gaussian process occupancy maps [4][5] can generate a continuous representation of the environment and capture the observation uncertainty inherent in the point cloud data. However, the uncertainty of the measurements themselves or the uncertainty of the robot locations from which the point cloud is captured is not addressed.

The main contribution of this paper is a technique for building a map together with its uncertainty from a point cloud. Uncertainty of each point in the point cloud is computed using sensor noise and the robot location uncertainties. The focus is to generate a compact, parameterised representation, where the mean and the covariance matrix of the parameter vector naturally captures the map uncertainty, exactly the same way as in the case of a geometric feature based map. In effect, these parameters can be thought of as high level features, making it possible to write appropriate observation equations relating these “features”, robot locations and sensor observations. As a result any existing probabilistic localisation technique may be used to generate robot location estimates that incorporate the uncertainty of

<sup>1</sup> J. Arukgoda, R. Ranasinghe and G. Dissanayake are with the Centre for Autonomous Systems, University of Technology Sydney, 15 Broadway, Ultimo NSW 2007, Australia. Email: {Janindu.Arukgoda, Ravindra.Ranasinghe, Gamini.Dissanayake}@uts.edu.au

the map. Furthermore, the possibility to extend these ideas for SLAM and motion planning under uncertainty also exist.

The key idea is to use a parametric approximation to the distance function (DF) to represent environments. Signed distance function (SDF) represents the magnitude of the distance to the closest occupied point in a given environment while the vector distance function (VDF) captures this magnitude as well as its direction. Both SDF and VDF are differentiable at the object boundaries unlike their unsigned counterpart making these suitable for the strategy proposed in this paper. Parametric approximation to the SDF/VDF using a set knot points of a cubic spline surface are obtained by enforcing the Eikonal equation over the region to be mapped and boundary conditions that captures the uncertainty of each of the points in the point cloud. The problem of obtaining the estimates of the coefficients and their uncertainties is formulated as a weighted non-linear least squares problem. Numerical examples of generating such parametric maps are presented to illustrate the proposed methodology.

This paper is organized as follows. In section II, a continuous parametric DF map representation that can capture the map error is presented. Section III explains the numerical examples with a discussion and conclusions in section IV.

## II. METHODOLOGY

This section outlines the proposed methodology. Properties of the distance function that can be used to capture all essential characteristics of a given environment are described first. A method to numerically solve the Eikonal equation at unit speed to obtain a parametric approximation to the distance function, followed by equations that are needed to compute the covariances associated with these parameters are then presented.

### A. Environment Representation With Distance Functions

Distance functions represent the distance at each location to the closest occupied space in a given environment. The geometry of the environment is implicitly captured in the DF as a level set of a function defined over the space in which the geometry is defined. Therefore, in addition to capturing the occupied regions of the environment, by providing a continuous measure of the distance to the closets occupied region, it offers a much richer representation in comparison to occupancy grid maps[6]. There are several types of distance functions. The unsigned distance function (UDF) provides the absolute distance to the closest occupied region while the signed distance function (SDF) is given a sign corresponding to whether the point under consideration is in free or occupied space. The vector distance function (VDF) captures both the magnitude of the distance and the direction to the closest occupied point in space. All three types of DF have been used to localise mobile robots[7][8][9][10]. However, at the boundaries, the gradients of UDF are discontinuous whereas they are continuous for SDF and VDF, making UDF unsuitable for localisation frameworks which use derivatives as locations used to compute the observation equation typically fall around this region. Although truncated

SDF is widely used to match point clouds[11][12], clearly defining regions as 'inside' or 'outside' in a complex environment is difficult, and sometimes impossible. VDF on the other hand overcomes this difficulty [10]. In the following, the problem formulation based on VDF is presented. The equations associated with SDF can easily be derived as SDF is a special case of VDF.

If  $u(x)$ ,  $x \in \mathbb{R}^2$  is the distance from point  $x$  to a manifold  $S$  where  $dim(S) = 0$  (i.e. a set of points), then  $u(x)$  describes the unsigned distance transform. It has been shown that UDF is also the entropy satisfying solution of an Eikonal equation at unit speed[13] shown in (1).

$$|\nabla u| = 1, u|_S = 0 \quad (1)$$

The relationship between a VDF, denoted by the function  $DF_v : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ , and a UDF, denoted by the function  $DF_u : \mathbb{R}^2 \rightarrow \mathbb{R}^1$  for any  $(x, y) \in \mathbb{R}$  is given by (2)

$$DF_u(x, y)^2 = DF_x(x, y)^2 + DF_y(x, y)^2 \quad (2)$$

where  $DF_x$  and  $DF_y$  are the two orthonormal components of  $DF_v$ [10].

From (1) and (2), a partial differential equation (PDE), for which the solution is a VDF can be derived as (3)

and the boundary conditions are given by (4).

$$DF_x(x, y) = 0, DF_y(x, y) = 0, \forall (x, y) \in S \quad (4)$$

### B. Parametric Approximation of a Solution to a PDE

Given a problem modeled as (5) and (6) where  $\mathcal{L}$  and  $\mathcal{B}$  are differential operators,  $f$  and  $g$  are given functions and  $\partial\Omega$  is the boundary of a bounded open domain  $\Omega$ , an approximate solution for a PDE can be derived by reducing the problem to an unconstrained optimization problem when a function with enough parameters that can be arbitrarily close to any element in the underlying space  $\mathcal{V}$  is available. Given such function  $u_a(x, \beta_i)$  where  $\beta_i$  represents the parameters, the objective function of the unconstrained optimization problem that results in the approximate solution  $u = u_a(x, \beta_i)$  is given by (7) [14].

$$\mathcal{L}u = f, x \in \Omega \quad (5)$$

$$\mathcal{B}u = g, x \in \partial\Omega \quad (6)$$

$$h = \int_{\Omega} \|\mathcal{L}u_a - f\|^2 d\mathcal{V} + \int_{\partial\Omega} \|\mathcal{B}u_a - g\|^2 dS \quad (7)$$

### C. Cubic Spline Surface Approximation of VDF

Cubic splines are widely used for representing complex shapes. A cubic spline surface is constructed by a set of surface patches that connect with  $C^2$  continuities. A surface patch can be defined as (8)

$$\frac{(DF_x \frac{\partial DF_x}{\partial x} + DF_y \frac{\partial DF_y}{\partial x})^2 + (DF_x \frac{\partial DF_x}{\partial y} + DF_y \frac{\partial DF_y}{\partial y})^2}{DF_x^2 + DF_y^2} = 1 \quad (8)$$

$$P(u, w) = \left(\frac{1}{6}\right)^2 [u^3, u^2, u, 1] M P M^T \begin{bmatrix} w^3 \\ w^2 \\ w \\ 1 \end{bmatrix} \quad (9)$$

where  $M$  is the basis matrix defined as (9),  $P$  is a 4x4 control point grid defined as (10) and  $0 \leq u, w \leq 1$  are the parameters that describe the relative location within the surface patch with four corner points  $K_{00} = P(0, 0)$ ,  $K_{01} = P(0, 1)$ ,  $K_{10} = P(1, 0)$ ,  $K_{11} = P(1, 1)$ . The set of control points  $P_{ij}$  is the sole set of parameters that govern the behavior of the cubic spline surface.

$$M = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \quad (9)$$

$$P = \begin{bmatrix} P_{00} & P_{01} & P_{02} & P_{03} \\ P_{10} & P_{11} & P_{12} & P_{13} \\ P_{20} & P_{21} & P_{22} & P_{23} \\ P_{30} & P_{31} & P_{32} & P_{33} \end{bmatrix} \quad (10)$$

It will be numerically illustrated in the next section that cubic splines provide a suitable approximation to the distance function given a sufficiently dense set of control points. Furthermore as they have local support, the jacobians associated with the optimisation problem defined by (7) become sparse. The objective function  $h_v$  for a VDF representation of the environment is given by (11). A similar equation can be derived in the case where the environment is to be represented by its SDF. It is important to note that the VDF and the gradient of the SDF are both discontinuous at the cut locus [6]. Therefore, the cubic spline approximation is not able to represent the distance function in this region and as a consequence Eikonal equation can not be enforced. However, previous work with the use of distance function representations for localisation [15] has shown that it is not essential to accurately represent the distance function near its discontinuities as in practice only a very small percentage of observations fall in these regions and therefore such observations can be discarded without significant information loss.

#### D. Estimating Map Uncertainty

This subsection presents the essential equations for generating the uncertainty associated with the map representation proposed in the specific situation where a point cloud is acquired from multiple locations in a two-dimensional environment using a robot equipped with a laser range finder.

Let  $\hat{\mathbf{x}}_{\mathbf{R}} = (x_r, y_r, \phi_r)^T$  be a mobile robot pose estimate in a 2D environment with covariance  $cov(\hat{\mathbf{x}}_{\mathbf{R}})$  and  $S_{r\theta} = \{(r_i, \theta_i)\}$  be the observations obtained from the laser range

finder with a range uncertainty of  $\sigma_r$  at  $\hat{\mathbf{x}}_{\mathbf{R}}$ . Each observation from the sensor can be projected into the world coordinate frame using (12) to obtain a point cloud.

$$X_{oi} = \begin{Bmatrix} x_{oi} \\ y_{oi} \end{Bmatrix} = \begin{Bmatrix} x_r + r_i \cos(\theta_i - \phi_r) \\ y_r + r_i \sin(\theta_i - \phi_r) \end{Bmatrix} \quad (12)$$

The uncertainties of the robot pose estimate and range measurements can be propagated to the each of the points in the point cloud  $cov(X_{oi})$  as (13).

The uncertainty of the observations can be integrated as a set of weights  $\rho_{oi}$ , derived as (14) to remodel objective function described in (11). Although the Eikonal equation described in (3) should be satisfied everywhere in  $\mathcal{V}$ , there will be a residue that depends on the quality of the approximation. Accordingly, a weight  $\rho_{ei}$  that is empirically determined using simulation examples needs to be associated with the relevant elements of the objective function. This leads to the weighted non linear least squares problem  $h_{v^*}$  defined in (15).

$$\rho_{oi} = \frac{1}{\left(\frac{\partial(DF_x^2 + DF_y^2)}{\partial X_{oi}}\right) cov(X_{oi}) \left(\frac{\partial(DF_x^2 + DF_y^2)}{\partial X_{oi}}\right)^T} \quad (14)$$

In case of the VDF, the approximate solution consists of two cubic spline surfaces that correspond to  $DF_x$  and  $DF_y$ . On the other hand SDF requires one cubic spline surface. As the parameters representing the environment is obtained through solving an optimisation problem, an explicit function relating the inputs to this solution does not exist. Therefore the covariance matrices for the optimized parameter vectors corresponding to  $DF_x$  and  $DF_y$ , denoted by  $Px_{ij}^*$  and  $Py_{ij}^*$  are given by (16) and (17)

$$cov(Px_{ij}^*) = J_x * cov(X_{oi}) * J_x^T \quad (16)$$

$$cov(Py_{ij}^*) = J_y * cov(X_{oi}) * J_y^T \quad (17)$$

where  $J_x$  and  $J_y$  are the corresponding Jacobians at the optimal solution.  $J_x$  is derived using the implicit function theorem[16] as (18) where  $H$  is the Hessian matrix of  $h_{v^*}$  with respect to  $Px_{ij}$ . It is straightforward to derive a similar equation for  $J_y$ .

$$J_x = -H^{-1} * \left[ \frac{\partial^2 h_{v^*}}{\partial Px_{ij}^* \partial r_k} \right] \quad (18)$$

### III. EXPERIMENTAL RESULTS

This section illustrates the proposed approach using two simulation experiments.

- 1) Experiment 1 : A robot equipped with a 180° FOV LiDAR observing an L-shaped environment from three poses is used to acquire a point cloud.

$$h_v = \int_{\Omega} \left\| \left( \frac{(DF_x \frac{\partial DF_x}{\partial x} + DF_y \frac{\partial DF_y}{\partial x})^2 + (DF_x \frac{\partial DF_x}{\partial y} + DF_y \frac{\partial DF_y}{\partial y})^2}{DF_x^2 + DF_y^2} \right) - 1 \right\|^2 d\mathcal{V} + \int_{\partial\Omega} (DF_x^2 + DF_y^2) dS \quad (11)$$

$$\text{cov}(X_{oi}) = \left( \frac{\partial X_{oi}}{\partial \hat{\mathbf{x}}_{\mathbf{R}}} \right) \text{cov}(\hat{\mathbf{x}}_{\mathbf{R}}) \left( \frac{\partial X_{oi}}{\partial \hat{\mathbf{x}}_{\mathbf{R}}} \right)^T + \left( \frac{\partial X_{oi}}{\partial r} \right) \sigma_r^2 \left( \frac{\partial X_{oi}}{\partial r} \right)^T \quad (13)$$

$$h_{v^*} = \int_{\Omega} \rho_{ei} \left\| \left( \frac{(DF_x \frac{\partial DF_x}{\partial x} + DF_y \frac{\partial DF_y}{\partial x})^2 + (DF_x \frac{\partial DF_x}{\partial y} + DF_y \frac{\partial DF_y}{\partial y})^2}{DF_x^2 + DF_y^2} \right) - 1 \right\|^2 d\mathcal{V} + \int_{\partial\Omega} \rho_{oi} (DF_x^2 + DF_y^2) dS \quad (15)$$

Measurement	Noise Parameter
Position Estimates	$\sigma_x = 0.01\text{m}$ $\sigma_y = 0.01\text{m}$
Orientation Estimates	$\sigma_{\phi} = 0.005\text{rad}$
Range Measurements	$\sigma_r = 0.02\text{m}$

TABLE I: Noise Parameters used in Experiment 1

- 2) Experiment 2 : A Turtlebot robot equipped with a 270° FOV LiDAR navigates through a corridor in an office environment simulated in Gazebo is used to gather the point cloud.

#### A. Experiment 1

A robot equipped with a 180° FOV LiDAR observing an L-shaped environment from 3 poses is simulated as shown in Fig. 2. The range observations and pose estimates are corrupted using zero mean Gaussian noise with the parameters given in Table I. The point cloud map generated from the resulting pose estimates and range measurements is shown in Fig. 3.

Given the simplicity of the environment, SDF was used as the basis for its representation where sign of the distance function was set to negative inside the room. To determine an appropriate control point density required to approximate the SDF for this environment, numerical experiments in which the true SDF is approximated with cubic spline surfaces with different resolutions were conducted. Fig. 4 demonstrates the difference between the true SDF and its least square approximation at different resolutions. Note that there are large errors in the regions near the cut locus where the gradient of the distance function is discontinuous and therefore does not satisfy the Eikonal equation as mentioned previously. Using this data as a basis, control point resolution of 0.2m was selected for the cubic spline approximation.

At the optimal solution, the zero contour of the SDF approximation is shown with the observed point cloud in Fig. 5. The approximated SDF is shown in Fig. 6. The values of the approximated SDF at the true boundary locations are shown in Fig. 7 with the associated two-sigma bounds. Fig. 8 shows the zero contour of the SDF at the optimal solution, approximated by multiple resolutions.

#### B. Experiment 2

A Turtlebot robot equipped with a 270° FOV LiDAR is simulated to navigate through a corridor in an example office

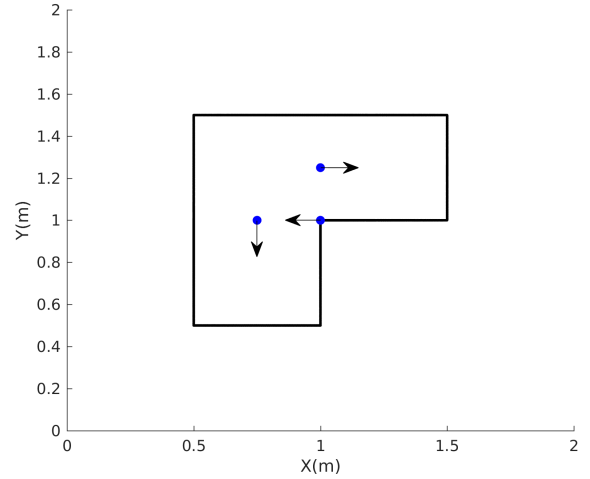


Fig. 2: Experiment 1 : Environment and Robot Poses

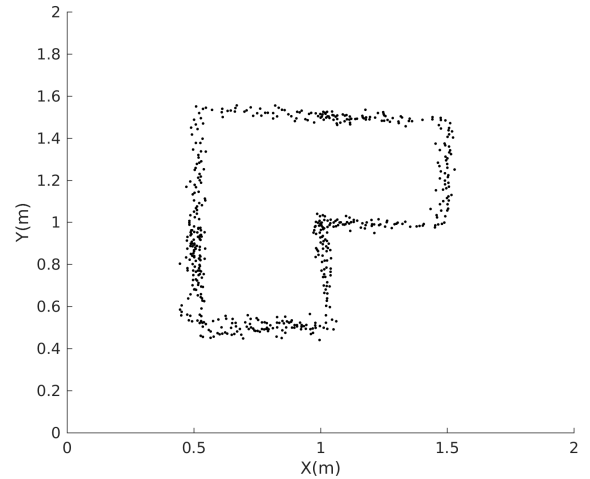


Fig. 3: Experiment 1 : The Point Cloud Obtained

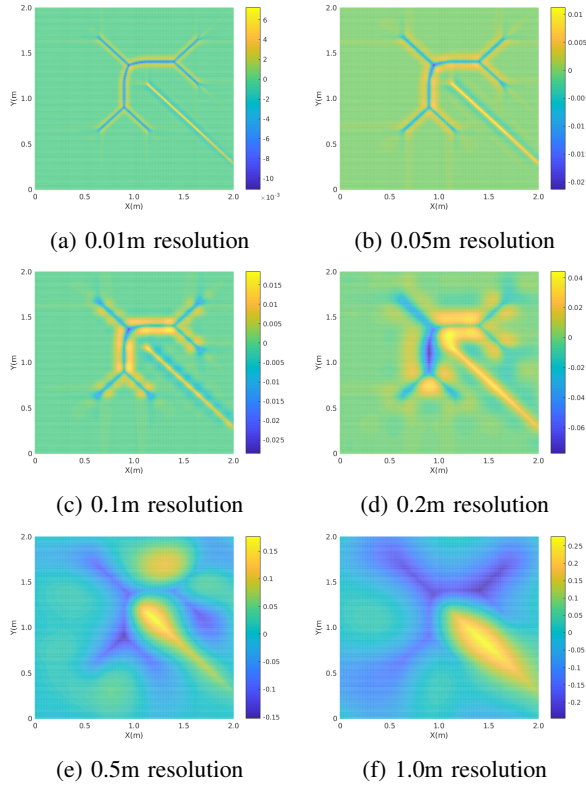


Fig. 4: Error (meters) in Approximating the SDF at Different Resolutions

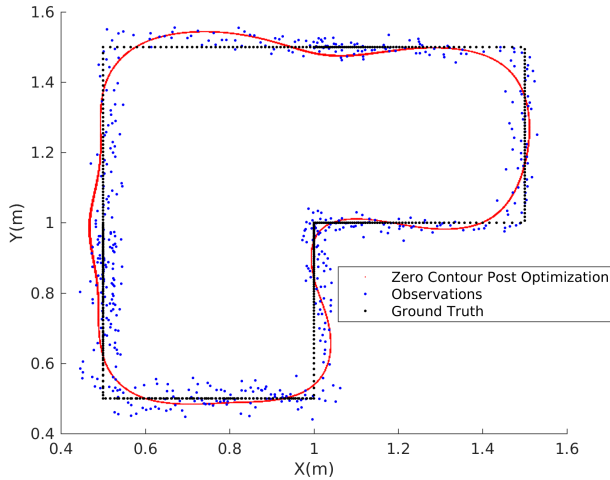


Fig. 5: Experiment 1 : Zero Contour of the SDF Representation, Observations and True Boundary

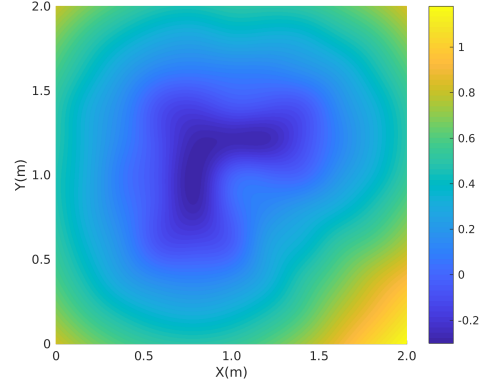


Fig. 6: Experiment 1: SDF Obtained Using The Derived Cubic Spline Coefficients

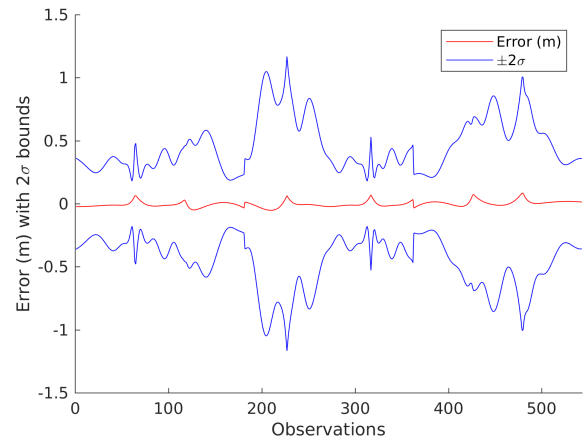


Fig. 7: Experiment 1: SDF and Its Uncertainty Bounds at the True Boundary

Measurement	Noise Parameter
Position Estimates	$\sigma_x = 0.05\text{m}$ $\sigma_y = 0.05\text{m}$
Orientation Estimates	$\sigma_\phi = 0.005\text{rad}$
Range Measurements	$\sigma_r = 0.02\text{m}$

TABLE II: Noise Parameters for Experiment 2

environment (Willow Garage office) in Gazebo. As in the case of experiment 1, the range measurements and the true robot pose are corrupted using zero mean Gaussian noise. The noise parameters are given in Table II. The point cloud map generated from the corrupted pose estimates and range measurements is shown in in Fig. 9. A VDF is approximated in this experiment.

To manage the computational load, the VDF is approximated with a control point resolution of 0.25m. Fig. 10 shows the resulting OGM at the same resolution.

#### IV. DISCUSSION AND CONCLUSIONS

The proposed method for representing environments using a parameter vector and its covariance matrix presented in this paper can be seen as an attempt to unify occupancy grid maps that capture environment geometry using aggregation

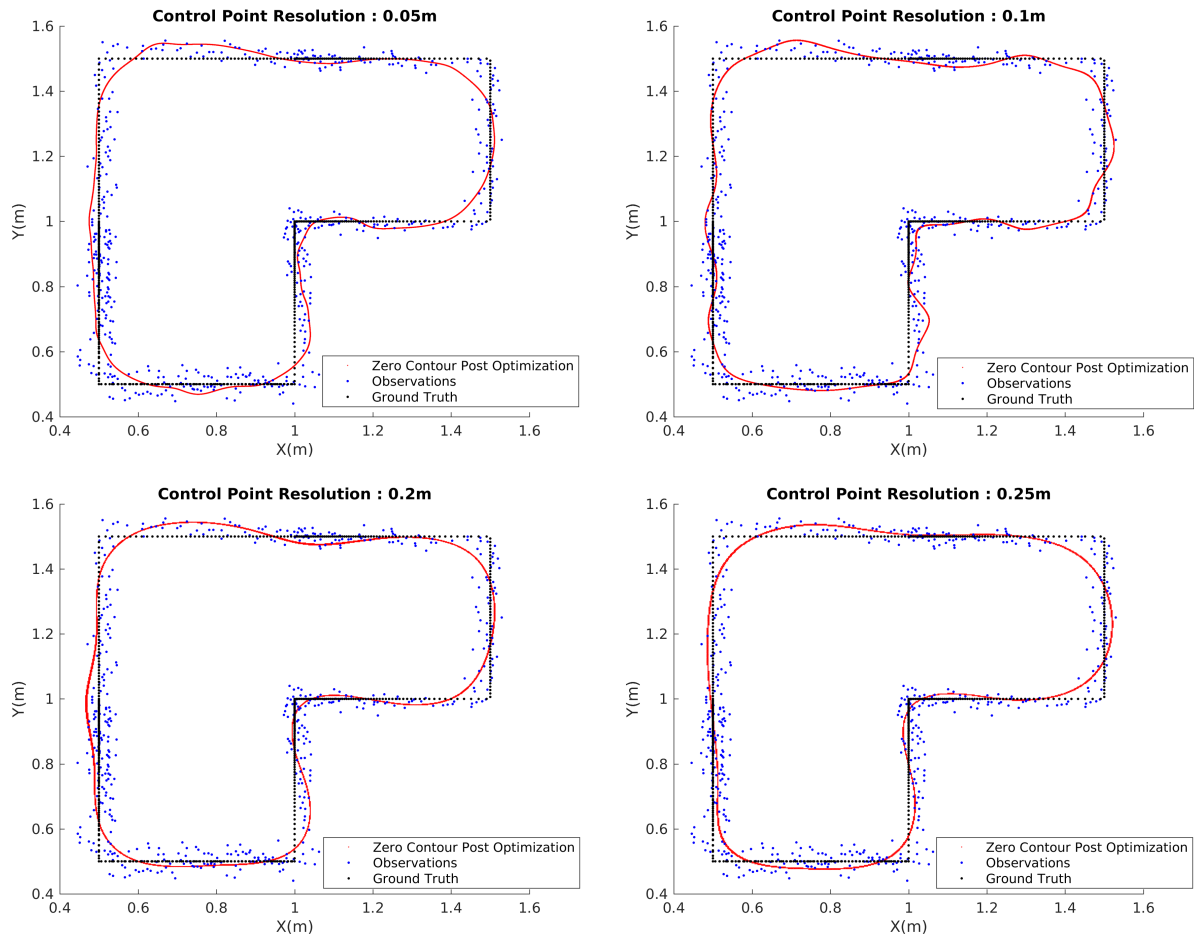


Fig. 8: Experiment 1 : Zero Contour of the SDF Representation at Different Resolutions, Observations and True Boundary

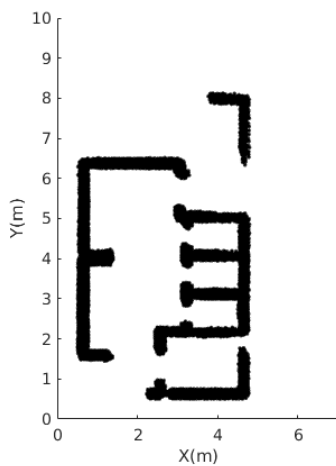


Fig. 9: Experiment 2 : Point Cloud Map

of point clouds with maps that are a collection geometric features such as points and lines. Ability to represent point clouds with a parameter vector can help extend probabilistic techniques that can only be used with feature maps to deal with point clouds. During the experimental evaluation, it

was observed that weights in the objective function of the weighted non linear least squares problem has a significant impact on the accuracy of the resulting map. In the examples presented, these were derived by trial and error. The set of weights  $\rho_{oi}$  were normalized into a range between 1.0-2.0 inclusive. The weight  $\rho_{ei}$  was set to 0.125. Strategies to automatically determine these values is currently being explored.

There are a number of future directions for investigation, some of which are currently being pursued. The proposed methodology is computationally heavy. One way to address this is to represent environments with submaps that are tractable and develop a strategy to merge these while preserving all the associated uncertainties. In the spirit of truncated signed distance functions that have become a popular in representing 3D point clouds, it is clearly worth investigating whether a non-uniform splines can be used to reduce the dimensionality of the problem without a significant loss of utility. Exploring whether there are a different basis functions that naturally satisfy the Eikonal equation in order to reduce the dimensionality of the representation to make the optimisation process converge faster and also provide guarantees that the true optimum has been reached

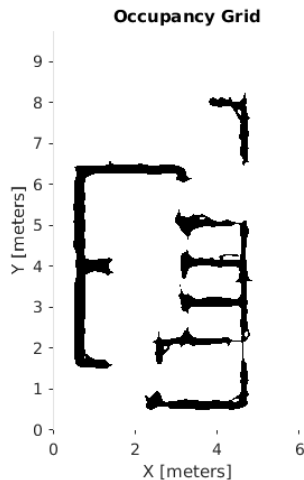


Fig. 10: Experiment 2 : OGM of the Office Space Corridor after Optimization

could be valuable. Using the parametrised representation and the associated covariance matrix in localisation and SLAM algorithms that are currently limited to environments with geometric features, for example extended Kalman filters, and evaluate their relative merits when compared with traditional solutions in occupancy grids is also likely to be a fruitful avenue for further research.

#### REFERENCES

- [1] G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (slam) problem," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 3, pp. 229–241, 2001.
- [2] M. Kaess, A. Ranganathan, and F. Dellaert, "isam: Incremental smoothing and mapping," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1365–1378, 2008.
- [3] V. Ila, J. M. Porta, and J. Andrade-Cetto, "Information-based compact pose slam," *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 78–93, 2010.
- [4] M. Ghaffari Jadidi, "Gaussian processes for information-theoretic robotic mapping and exploration," Ph.D. dissertation, 2017.
- [5] S. Vasudevan, F. Ramos, E. Nettleton, and H. Durrant-Whyte, "Gaussian process modeling of large-scale terrain," *Journal of Field Robotics*, vol. 26, no. 10, pp. 812–840, 2009.
- [6] R. Ranasinghe, G. Dissanayake, T. Furukawa, J. Arukgoda, and L. Dantamarayana, "Environment representation for mobile robot localisation," in *2017 IEEE International Conference on Industrial and Information Systems (ICIIIS)*, Dec 2017, pp. 1–6.
- [7] L. Dantamarayana, G. Dissanayake, and R. Ranasinghe, "C-log: A chamfer distance based algorithm for localisation in occupancy grid-maps," *CAAI Transactions on Intelligence Technology*, vol. 1, pp. 272–284, Oct. 2016.
- [8] M. Torres-Torriti and A. Guesalaga, "Scan-to-map matching using the hausdorff distance for robust mobile robot localization," in *2008 IEEE International Conference on Robotics and Automation*, May 2008, pp. 455–460.
- [9] F. Donoso-Aguirre, J.-P. Bustos-Salas, M. Torres-Torriti, and A. Guesalaga, "Mobile robot localization using the hausdorff distance," *Robotica*, vol. 26, no. 2, pp. 129–141, 2008.
- [10] J. Arukgoda, R. Ranasinghe, L. Dantamarayana, G. Dissanayake, and T. Furukawa, "Vector distance function based map representation for robot localisation," in *Australasian Conference on Robotics and Automation*, 2018.
- [11] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison *et al.*, "Kinectfusion: real-time 3d reconstruction and interaction using a moving depth

- camera," in *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*. ACM, 2011, pp. 559–568.
- [12] R. A. Newcombe, D. Fox, and S. M. Seitz, "Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.
- [13] S. Mauch, "A fast algorithm for computing the closest point and distance transform," *Go online to <http://www.acm.caltech.edu/seanm/software/cpt/cpt.pdf>*, 2000.
- [14] G. Dissanayake and N. Phan-Thien, "Neural-network-based approximations for solving partial differential equations," *communications in Numerical Methods in Engineering*, vol. 10, no. 3, pp. 195–201, 1994.
- [15] L. Dantamarayana, R. Ranasinghe, and G. Dissanayake, "C-log: A chamfer distance based method for localisation in occupancy grid-maps," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2013, pp. 376–381.
- [16] J. C. Clarke, "Modelling uncertainty: A primer," *Tutorial of Department of Eng. Science*, pp. 1–21, 1998.