Log-GPIS-MOP: A Unified Representation for Mapping, Odometry and Planning

Lan Wu, Ki Myung Brian Lee, Cedric Le Gentil, and Teresa Vidal-Calleja

***Abstract*—Whereas dedicated scene representations are required for each different task in conventional robotic systems, this paper demonstrates that a unified representation can be used directly for multiple key tasks. We propose the Log-Gaussian Process Implicit Surface for Mapping, Odometry and Planning (Log-GPIS-MOP): a probabilistic framework for surface reconstruction, localisation and navigation based on a unified representation. Our framework applies a logarithmic transformation to a Gaussian Process Implicit Surface (GPIS) formulation to recover a global representation that accurately captures the Euclidean distance field with gradients and, at the same time, the implicit surface. By directly estimating the distance field and its gradient through Log-GPIS inference, the proposed incremental odometry technique computes the optimal alignment of an incoming frame and fuses it globally to produce a map. Concurrently, an optimisation-based planner computes a safe collision-free path using the same Log-GPIS surface representation. We validate the proposed framework on simulated and real datasets in 2D and 3D and benchmark against the state-of-the-art approaches. Our experiments show that Log-GPIS-MOP produces competitive results in sequential odometry, surface mapping and obstacle avoidance.**

***Index Terms*—Gaussian Process Implicit Surfaces, Euclidean Distance Field, Mapping, Localisation, Planning, SLAM.**

I. Introduction

Simultaneous localisation and mapping (SLAM) is a well-known problem in robotics [1], which concerns reconstructing or building a map of an unknown environment while simultaneously tracking the location of a robot-sensor system within the map. Using the built map and the estimated location, a path planner may produce collision-free paths for the robot to navigate safely. Each different task raises varying requirements leading to dedicated environment representations. The choice of scene representation strongly affects the performance of the localisation, mapping, and path planning tasks.

For instance, motion estimation favours sparse representations such as the locations of features in the environment, which allows consistent estimation of the robot's location. A key objective of the mapping task is to reconstruct an accurate, dense and high-resolution map of the scene for, e.g., inspection purposes. Path planning similarly requires dense information such as obstacle occupancy or closest distance to collision in order to avoid obstacles. As a unified representation for multiple purposes, a feature-based representation such as a set of landmarks [3], [4] is able to reconstruct a sparse map along with self-localisation and path planning in an unknown environment. Although sparse feature-based representations have been widely used for representing geometric structures and have proven advantageous for localisation [3], [5], they are not suitable for an accurate depiction of the environment or path planning, where dense maps are more appropriate, although difficult to recover from sparse features. As an alternative, several approaches [6], [7] employ signed distance functions (SDF) as a representation for path planning and dense mapping. SDF can be used directly by a planner and the implicit surfaces can be used to recover the dense map. However, without matching sparse features to identify correspondences, it is difficult to perform a consistent alignment for motion

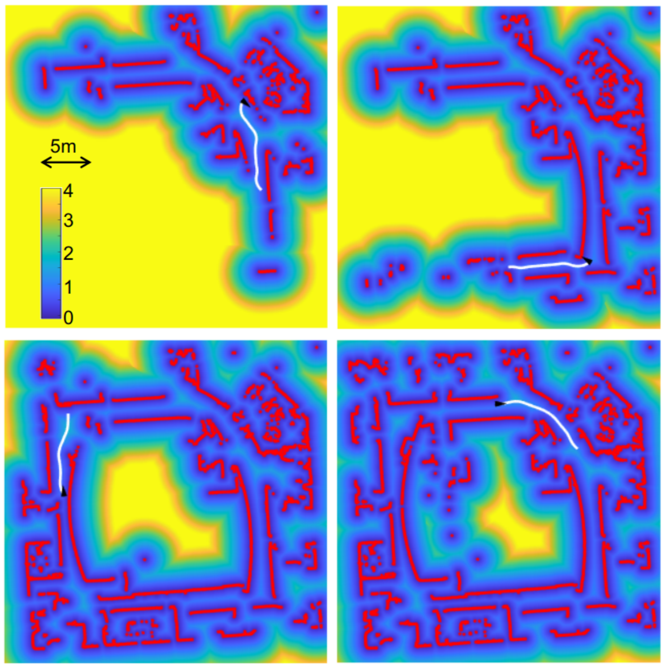


Fig. 1. The proposed framework Log-GPIS-MOP incrementally builds the distance field (yellow to blue colour map) and estimates the implicit surface (red) of the first loop around the Intel Lab [2]. Our incremental odometry and planning use the distance and gradients of the Log-GPIS as input to compute the current alignment and the white optimal trajectory for the robot to avoid colliding with the surfaces respectively.

This work was supported by the Australian Government Research Training Program. All authors are with the UTS Robotics Institute, Faculty of Engineering and IT, University of Technology Sydney, NSW 2007, Australia. (Corresponding author: Lan Wu.) `Lan.Wu-2@uts.edu.au`

This article and the supplementary downloadable material are available at https://ieeexplore.ieee.org, provided by the authors. The material consists of a video. This video does not aim to describe the method of the proposed approach, but it shows the experimental results of the proposed method.

estimation for localisation using the SDF representation [8].

In this work, we propose a unified representation so-called Log-GPIS suitable for localisation, mapping, and path planning (see Fig. 1), inspired by recent work on SDF representations with Gaussian process implicit surfaces (GPIS). GPIS [9], [10], [11] is a continuous and probabilistic representation that exploits the implicit surfaces. Whereas conventional GPIS representations [12], [13] can only infer accurately the Euclidean distance field (EDF) close to the measurements (noisy points on the surface), Log-GPIS, proposed in our previous work [14], faithfully models the EDF and gradients by enforcing an approximation of the Eikonal equation [15] through a simple log transformation on the standard GPIS. This representation allows us to use Log-GPIS as a single unified representation for localisation, mapping and planning multiple robotics tasks, which is the focus of this paper. The smooth and probabilistic nature of Log-GPIS provides an accurate dense surface for mapping given noisy measurements. The extrapolative power of Log-GPIS provides information not only near but also far away from the measurements, and hence enables estimation of the full EDF with gradients. This allows motion estimation via direct optimisation of the alignment between the global map representation and a new set of measurements. Furthermore, the EDF and its gradients are suitable for trajectory optimisation-based planners.

The contributions of this paper are as follows:

- The Log-GPIS-MOP framework, which achieves incremental localisation, mapping and planning based on a single continuous and probabilistic representation through accurate prediction of the EDF and the implicit surface with gradients.
- A sound formulation for 2D/3D point cloud alignment for odometry estimation, where the Log-GPIS is used to minimise the point cloud distance to the implicit surface from a set of newly arrived observations following the gradients.
- A generic mapping approach for organised and unorganised point clouds that consists in probabilistically fusing new measurements into a global Log-GPIS representation and is ready for a modified version of marching cubes to recover dense surface reconstruction.
- A path planning approach that uses Log-GPIS to avoid collisions with the mapped surfaces following the EDF and gradient information.

Moreover, we present a thorough evaluation of the overall Log-GPIS-MOP framework and individual tasks on public benchmarks, and qualitative and quantitative benchmarking with the state-of-the-art. Our work enables sequential and concurrent localisation, mapping and planning, focusing on accurate performance as opposed to real-time operations.

The paper structure is organised as follows. The related work is presented in Sec. II. Sec. III explains the background knowledge. Our Log-GPIS representation is discussed in Sec. IV. The methodology of the Log-GPIS-MOP framework is presented in Sec. V , with odometry in VI, mapping VII, and planning VIII. In Sec. IX, we present a thorough evaluation of the framework on public benchmarks and compare the performance to existing solutions. Finally, the conclusion and suggestions on future work are given in Sec. X.

II. Related Work

For visual SLAM, several successful frameworks process the input data into a set of sparse features as the representation for odometry and mapping. Recent feature-based SLAM systems [3], [16], [5] have demonstrated accurate pose estimation, consistency throughout long-term operations, and real-time performance. Due to the difficulties of extracting continuous surfaces from a sparse set of points, most of them are not suitable for applications other than localisation. In addition, dedicated and sophisticated algorithms are required to recover a dense and accurate reconstruction from sparse points [17].

As an alternative representation to achieve sensor tracking and mapping, the seminal work of KinectFusion [6] demonstrates high-performance camera motion estimation and dense reconstruction in real-time. To represent the geometry, KinectFusion uses the truncated signed distance function (TSDF) [18] and a coarse-to-fine ICP algorithm [19] to perform an alignment. TSDF, originally proposed in computer graphics literature, has become a widely used representation in odometry and mapping approaches. However, KinectFusion has some drawbacks. First, the algorithm implementation relies on GPU. Second, the distance of the TSDF is truncated and generally overestimates the distance within limited viewpoints. Third, the representation can not be directly used for planning.

Another relevant representation for mapping and odometry tracking is from RGB-D SLAM [20]. Such as in the framework recently proposed in [21], ElasticFusion represents the map as a collection of surfels, which are small surface elements with normals and colours. Given a surfel-based representation, ElasticFusion incrementally estimates the sensor pose through dense frame-to-map tracking. No pose-graph optimisation or post-processing is required. Such representation generally assists in computing accurate pose estimation, along with high-quality reconstruction and a rich understanding of the occupied space. However, the representation does not differentiate free space from unknown space explicitly, which is of limited applicability for autonomous navigation tasks.

Semantic combined with geometric representations are a recent development gaining popularity for mapping [22] and localisation [23]. The application has since been further extended for navigation [24]. It provides competitive efficiency, accuracy and robustness for metric-semantic SLAM and perception. However, the mesh representation used in these works does not provide inherently distance and direction to the collision which has to be computed separately for navigation.

Other representations based on explicit occupancy [25] efficiently compute and store information, such as free space, occupied space, and unknown. Occupancy maps are commonly used for mapping and path-planning applications. Many options for occupancy maps have been presented in the literature varying from discrete representations [25] [26] [27] to probabilistic and continuous ones [28], [29]. Discrete occupancy maps comprise a set of grid cells with occupancy information regarding the environment. Planning algorithms frequently take such discrete occupancy grids as input [30]. As for the probabilistic and continuous options, the authors in [31] propose continuous occupancy mapping by viewing Gaussian processes as implicit functions of occupancy. Doing so, it is capable of building the representation with both occupancy information and implicit surfaces. Extension of occupancy grids from 2D to 3D is, however, challenging, due to large memory usage. It is often difficult to perform fast ray-casting and look-up tabling for any space larger than a room. To this end, an online solution commonly used in 3D is Octomap [26]. This approach uses an octree-based structure to represent the occupancy probabilities of cells. The octree structure allows handling large spaces by immensely decreasing the amount of memory required to represent unknown or free space.

However, the representation with occupancy information alone is often insufficient for planning approaches. For instance, trajectory optimisation-based planners including CHOMP [7] and TrajOpt [32], require information on distance and direction to obstacles. In addition, having a distance map speeds up collision checking of complex environments. For example, robot arms are commonly represented as a set of overlapping spheres, in which case, it is much more convenient to query the distance field at the centre of each sphere for collision checking [7].

Such optimisation-based planners require a complete distance field that is not truncated further from the surface. For this reason, a Euclidean signed distance field (ESDF) containing distance values over the entire space is necessary. An ESDF can be naturally passed to a collision-checking algorithm to produce collision-free trajectories [7] [32]. Gradients can be computed using the distance values of the neighbouring cells, which allows CHOMP and other planners to follow the gradient direction of the distance to push the trajectory away from collision [33].

Following the above requirements, Oleynikova *et al* [34] proposed the Voxblox framework to incrementally generate a representation for mapping and planning. It adopts the advantages of TSDF to generate an implicit surface, as its denseness is ideal for human visualisation. When using TSDF, the distance within the truncated threshold is relatively accurate. Then, an ESDF representation is numerically approximated from the TSDF to cover the observations' fields of view. The basic unit for ESDF is a voxel grid, which contains the distance to the nearest obstacles and the gradient thereof. The ESDF can be used with trajectory optimisation methods for online navigation, such as the one in [35] proposed by the same authors. Similarly, FIESTA [36] proposes a discrete voxel grid representation with SDF for mapping and planning, which reported comparable results to [37].

Compared to KinectFusion, Voxblox does not use the distance information from the TSDF or ESDF to perform motion estimation. The system still requires ground truth poses as input from a motion-capturing system or estimated poses from a visual-inertial odometry framework [38]. To make the distance play a role in the trajectory estimation, the same authors extended their work to tackle the long-term consistent mapping problem in Voxgraph [8]. It introduces submap-based ESDF and TSDF representations for mapping and navigation. Voxgraph adopts the distance information to generate correspondence-free constraints between submaps. Then, through optimising a pose graph, the relative pose given submaps in a closed loop is computed to mitigate long-term drifts. However, Voxgraph still requires external visual odometry to produce submaps from the incoming sensor data for further optimisation and localisation.

An interesting recent development is the use of neural networks for continuous implicit representation for reconstruction and mapping [39], [40], [41], [42]. Neural representations can be trained to model radiance fields [41], signed distance functions [39], [40], [43] or occupancy maps [42]. Most prominent is the neural radiance field (NeRF) [41], which represents the 3D environment with a continuous volumetric density function in tandem with view-dependent colours, both modelled by a neural network. Such neural representations have shown great promise for SLAM and localisation [44], [45], [46] as well as path planning [37], [47], [48], [49]. iMAP [44] first proposed to embed NeRF within a SLAM framework, followed by NICE-SLAM [46], which uses multiple NeRFs organised into a voxel grid. Using the constructed NeRF, trajectory planners have been proposed to optimise the path to avoid obstacles, which can be identified in NeRFs as regions of high density [48], [49]. Of those, [49] shows that planning performance can be improved by using an approximate ESDF derived from a NeRF. Our work learns an EDF, an unsigned variant of ESDF, which facilitates all of localisation, mapping and planning. Further, although these neural representations also serve as a sound representation for multiple tasks, we achieve similar functionalities with GPs, which we believe is significantly more 'white-box' than a neural network, and hence easier to understand and diagnose for practitioners. This comes with the added benefit of explicit uncertainty representation, which is not readily available with neural network models.

Furthermore, learning SDFs directly with neural network models has also gained popularity in recent years [39], [40], [43], which has been also extended to mapping and navigation [37]. A recent work closely related to ours is [39], which reports improvement in model performance through the use of a penalty to account for the Eikonal equation. The penalty is incorporated into the loss function, and the Eikonal equation is thus enforced via multiple gradient updates. In contrast, our framework elegantly incorporates a regularised version of the Eikonal equation through a single log transformation to accurately approximate the true EDF.

As mentioned above our representation is based on Gaussian Process implicit surfaces (GPIS), a continuous and probabilistic representation that was initially proposed to reconstruct surfaces given noisy surface observations [9]. Later in [12] and [50], the authors used the GPIS as an approximate signed distance function (GPIS-SDF) that infers the distance to the nearest surfaces. Based on the distance values, a localisation solution is formulated to find consecutive poses in [50]. However, it only keeps track of the distance information close to the surface as there is a lack of observations further away. The localisation requires the Kalman filter to provide the initial guess and odometry is performed with frame-to-map registration but does not leverage a pose graph post-processing step as proposed in here to further improve the estimates' accuracy. Based on the Varadhan's approach [51], our previous work in [14] applies the logarithmic transformation to a GPIS with a specific covariance function to recover the distance field in the space. This method provides the extra potential for GPIS to accomplish accurate distance field mapping even further away from the surface. However, our previous work does not consider solving the localisation problem as it assumes the sensor poses are known and do not actually propose a planning approach that leverages such a representation.

Through the related works, catering for the various requirements to have a representation for multiple purposes is still an open research problem. Thus, in the paper, we present the unified environment representation that can be used for dense reconstruction, collision avoidance, while providing additional information needed for the sensor's odometry estimation.

III. Background

In this section, we introduce the Eikonal equation, the derivation of the distance approximation used in this work and a brief description of the Gaussian Process Implicit Surfaces.

A. Euclidean Distance Field

We assume that a robot's given environment can be modelled as a bounded manifold denoted as $S \subset \mathbb{R}^D$ with a suitably smooth boundary ∂S. The boundary ∂S is assumed to be orientable, which means it has a continuously varying surface normal at every point. We seek to represent the boundary ∂S in terms of its EDF $d(\mathbf{x})$, which is the nearest distance between a given point $\mathbf{x} \in \mathbb{R}^D$ and points on the boundary $\mathbf{w} \in \partial S$:

$$d(\mathbf{x}) = \min_{\mathbf{w} \in \partial S} |\mathbf{x} - \mathbf{w}| . \tag{1}$$

The EDF (1) represents the boundary ∂S implicitly as its zero-level set.

A definitive property of EDFs is that it satisfies the *Eikonal equation* almost everywhere[1], which states that the gradient is of unit norm:

$$|\nabla d(\mathbf{x})| = 1. \tag{2}$$

The intuition behind the Eikonal equation is that the fastest increase in the EDF $d(\mathbf{x})$ with a unit change in $\mathbf{x}$ is a unit change in distance, which occurs when moving in the normal direction to the surface. Since the gradient $\nabla d(\mathbf{x})$ represents the direction and rate of fastest increase in $d(\mathbf{x})$, it has a unit magnitude and is directed normally to the surface. Our framework represents the EDF *faithfully* by closely approximating the Eikonal equation (2) through a simple log transformation.

[1]Unsigned EDFs are not differentiable at the boundary ∂S, and hence the Eikonal equation does not hold at ∂S.

B. Heat-based distance function

In the robotics literature, recently proposed methods such as [34] [52] adopted the concept of wavefront propagation to estimate the distance field efficiently. This approach incrementally propagates the distance field to the neighbours directly from TSDF or from the occupancy map through a voxel grid. In this way, a compromise has to be made to not obtain the EDF directly as further post-processing algorithm is required. To recover the distance field exactly, one option is to solve or approximate the Eikonal equation. However, the main restriction in exploiting the Eikonal equation (2) is that the equation is hyperbolic and non-linear, which makes it difficult to directly solve and recover the accurate distance values.

Inspired by the idea proposed in the computer graphics literature [15], we motivate our work based on the physical phenomenon of heat propagation in space. The main idea is that the amount of heat propagated from a heated surface for infinitesimal time reflects the closest distance to the surface.

Let us consider the surface ∂S to be heated at an arbitrary temperature of one. Formally, the heat equation, the boundary conditions, and initial conditions are as follows:

$$\frac{\partial v(\mathbf{x},t)}{\partial t} = \Delta v(\mathbf{x},t) \text{ with } \begin{cases} v(\mathbf{x},t) = 1 \text{ when } \mathbf{x} \in \partial S \\ v(\mathbf{x},0) = 0 \text{ when } \mathbf{x} \notin \partial S, \end{cases} \tag{3}$$

where Δ is the Laplacian $\sum_{i=0}^{D} \frac{\partial^2}{\partial x_i^2}$, and $v(\mathbf{x},t)$ is the heat at location $\mathbf{x} \in \mathbb{R}^D$ and time t. For a small time period around $t = 0$, the heat equation (3) can be approximated as a screened Poisson equation[2]:

$$v(\mathbf{x},t) = t \Delta v(\mathbf{x},t). \tag{4}$$

The celebrated result of Varadhan [51, Theorem 2.3] is that the solution to (4) is related to the EDF as follows:[3]

$$d(\mathbf{x}) = \lim_{t \to 0} \{-\sqrt{t} \ln(v(\mathbf{x},t))\}. \tag{5}$$

Intuitively, (5) states that the negative logarithm of the heat propagated from a surface after a small quantum of time is proportional to the distance to the surface.

Importantly, Varadhan's formula (5) enables approximating the EDF as $d(\mathbf{x}) \approx -\sqrt{t} \ln(v(\mathbf{x},t))$ for a sufficiently small t. The Eikonal equation (2) remains approximately satisfied by this approximate EDF [53]. Namely, substituting $v(\mathbf{x},t) \approx \exp\left(-\frac{d(\mathbf{x})}{\sqrt{t}}\right)$ into (4) recovers a regularised form of Eikonal equation (2):

$$1 = |\nabla d(\mathbf{x})|^2 - \sqrt{t} \Delta d(\mathbf{x}), \tag{6}$$

where $\sqrt{t}\Delta d(\mathbf{x})$ is the regulariser. Accordingly, it is clear that the heat-based approximation (5) improves as the time t decreases.

[2]This is by discretising the temporal differentiation using the fact that $\frac{\partial v(\mathbf{x},t)}{\partial t}|_{t=0} = \lim_{t \to 0} \frac{v(\mathbf{x},t) - v(\mathbf{x},0)}{t}$ and $v(\mathbf{x},0) = 0$, which leads to $\frac{\partial v(\mathbf{x},t)}{\partial t}|_{t=0} = \lim_{t \to 0} \frac{v(\mathbf{x},t)}{t}$.

[3]Varadhan's original theorem only applies to points in S [51], but we proposed to extend it to all points in $\mathbb{R}^D$ by applying the theorem again to the complement S^c (i.e. outside S) [14].

C. Gaussian Process Implicit Surfaces

Gaussian Process (GP) [54] is a non-parametric stochastic method ideal for solving non-linear regression problems. GPIS techniques [9], [10], [11], [55] use a GP approach to estimate a probabilistic and continuous representation of the implicit surface given noisy measurements. Further proposed in [56], [12], [13], GPIS representation can be used for distance field estimation, which provides approximate distance values only near the surface.

The following provides the background knowledge of GPIS formulation for implicit surface and distance estimation. GP is capable of modelling the unknown function with gradients because the derivative of a GP is another GP [54], [11]. In the GPIS formulation, considering the unknown function, d represents the distance to a surface and ∇d denotes the corresponding gradient of d. Then d with ∇d can be modelled by the joint GP with zero mean:

$$\begin{bmatrix} d \\ \nabla d \end{bmatrix} \sim \mathcal{GP}(0, \tilde{k}(\mathbf{x}, \mathbf{x}')). \tag{7}$$

and covariance matrix

$$\tilde{k}(\mathbf{x}, \mathbf{x}') = \begin{bmatrix} k(\mathbf{x}, \mathbf{x}') & k(\mathbf{x}, \mathbf{x}') \nabla_{\mathbf{x}'}^{\top} \\ \nabla_{\mathbf{x}} k(\mathbf{x}, \mathbf{x}') & \nabla_{\mathbf{x}} k(\mathbf{x}, \mathbf{x}') \nabla_{\mathbf{x}'}^{\top} \end{bmatrix}, \tag{8}$$

where $k(\mathbf{x}, \mathbf{x}')$ is the covariance function of the function d. $\nabla_{\mathbf{x}}$ and $\nabla_{\mathbf{x}'}$ are the partial derivatives of $k(\mathbf{x}, \mathbf{x}')$ at $\mathbf{x}$ and $\mathbf{x}'$ [11] [54].

Let us consider a set of points and the measurements of points corrupted by noise. The distance measurements $\mathbf{y} = \{y_j = d(\mathbf{x}_j) + \epsilon_j\}_{j=0}^{J} \subset \mathbb{R}$ and the corresponding gradients $\nabla \mathbf{y} \subset \mathbb{R}^D$ are taken at locations $\mathbf{x}_j \in \mathbb{R}^D$ and affected by additive Gaussian noise $\epsilon_j \sim \mathcal{N}(0, \sigma_d^2)$. The set of point positions, the measurements and the corresponding gradients are the input training data. The posterior distribution of d at an arbitrary testing point $\mathbf{x}_*$ is given by $d(\mathbf{x}_*) \sim \mathcal{N}(\bar{d}_*, \mathbb{V}[d_*])$ with the predictive mean and variance with derivatives are expressed as follows:

$$\begin{bmatrix} \bar{d}_* \\ \nabla \bar{d}_* \end{bmatrix} = \tilde{\mathbf{k}}_*^{\top} (\tilde{K} + \sigma_d^2 I)^{-1} \begin{bmatrix} \mathbf{y} \\ \nabla \mathbf{y} \end{bmatrix} \tag{9}$$

$$\begin{bmatrix} \mathbb{V}[d_*] \\ \mathbb{V}[\nabla d_*] \end{bmatrix} = \tilde{k}(\mathbf{x}_*, \mathbf{x}_*) - \tilde{\mathbf{k}}_*^{\top} (\tilde{K} + \sigma_d^2 I)^{-1} \tilde{\mathbf{k}}_*. \tag{10}$$

Note that I represents the identity matrix of size $J(D+1) \times J(D+1)$. $\tilde{\mathbf{k}}_*$ represents the covariance vector between the training points and the testing point $\mathbf{x}_*$. $\tilde{K}$ is the covariance matrix of the training points J with gradients. Furthermore, $\tilde{k}(\mathbf{x}_*, \mathbf{x}_*)$ is the covariance function between the testing point.

Most GPIS techniques ensure the accuracy of the reconstructed surfaces, but only a few of them directly model the distance field [12], [13]. However, one of the disadvantages of standard GPIS applying inference of the distance field is that only accurately infers the distance near the surface. Given the limited training data, it is hard to keep the high precision and continuity of the distance field further from the measurements. To overcome this shortcoming, we introduce the Log-GPIS representation in Sec. IV. Based on the physics of heat propagation, Log-GPIS leverages both the well-known GPIS formulation and Varadhan's results to approximate the Euclidean distance field over the full space $\mathbb{R}^D$ given minimal and sparse observations close to the surface. With accurate distance predicted, we unlock the potential of Log-GPIS to accomplish multiple robotic tasks. Other than mapping in Sec. VII, we will discuss the Log-GPIS based odometry in Sec. VI and path planning in Sec. VIII.

IV. Log-Gaussian Process Implicit Surfaces

In this section, we introduce the derivation of the Log-GPIS representation and the choice of covariance function suitable for Log-GPIS with gradient inferences.

A. Derivation

Our Log-GPIS representation is built upon the heat-based distance approximation presented in Section III-B. We use a GP to represent the heat $v(\mathbf{x})$ and its gradient $\nabla v(\mathbf{x})$ after a quantum of time t:

$$\begin{bmatrix} v \\ \nabla v \end{bmatrix} \sim \mathcal{GP}(0, \tilde{k}(\mathbf{x}, \mathbf{x}')). \tag{11}$$

The function $v(\mathbf{x})$ must be a solution of the screened Poisson PDE (4). This solution can be enforced by choosing a specific kernel function $\tilde{k}$. We discuss the kernel choice in the following subsection.

Assuming that the choice of the covariance function respects the screened Poisson equation, we estimate $\bar{v}(\mathbf{x})$, $\nabla \bar{v}$ and the corresponding variances by simply using the regression equations (9) and (10). By applying (5), we recover the distance field $d(\mathbf{x})$ as:

$$\bar{d}_* = -\sqrt{t} \ln \bar{v}_*. \tag{12}$$

This logarithm transformation gave its name to our method, *Log*-GPIS. To ensure that $\bar{v}_*$ is positive, we use its absolute value in our implementation. The set of surface observations $\mathbf{y}$ in (9) corresponds to virtual heat observations on the surface S. In other words, the vector $\mathbf{y}$ is equal to $\mathbf{1}$ (cf. boundary conditions in (3)). We also use the surface normals, equivalent to the normalised gradient of the implicit distance function to compute the gradient observations $\nabla \mathbf{y}$.

Despite its simplicity, the log transformation has a non-trivial benefit as the predicted distance will approach infinity when querying points further away from the surface unlike the standard GPIS, which will predict a distance value of zero (falling back to the mean far away from the measurements). In other words, Log-GPIS avoids the undesirable artifacts further away from the predicted surface. One shortcoming of the proposed method is that it does not predict whether $\mathbf{x}$ is inside or outside the surface while standard GPIS does (i.e., Log-GPIS predicts EDF instead of SDF). [4] In our perspective, given the advantage of using (4) to approximate the distance field everywhere on $\mathbb{R}^D$, losing the sign is a small price to

[4]This follows from the extension of Varadhan's formula (5) from S to $\mathbb{R}^D$ (see footnote 3). Applying Varadhan's formula on both S and the complement S^c implies 'stitching' the EDFs of S and S^c, and hence whether a point belongs to S is no longer distinguishable.

pay for the convenience of the simplicity and accuracy of the proposed distance estimation.

We also observe that the log transformation affects the gradient. Taking the partial derivative of (12) with respect to the query point $\mathbf{x}_*$ leads to:

$$\nabla \bar{d}_* = \frac{-\sqrt{t}}{\bar{v}_*} \nabla \bar{v}_*, \tag{13}$$

where we can find that the main difference between $\nabla \bar{d}_*$ and $\nabla \bar{v}_*$ is the direction and the scaling factor $\sqrt{t}/\bar{v}_*$. Typically, for a standard GPIS, the magnitude of the gradients at the surface does not have to be fixed. However, the Eikonal equation (2) requires that the magnitude of the gradient is normalised to 1. Since the gradient of $\bar{d}_*$ and $\bar{v}_*$ have opposite directions, we simply normalise the gradient of $\bar{v}_*$, and invert the sign to obtain $\nabla \bar{d}_*$.

Based on (13), the predictive variance $\mathbb{V}[d_*]$ can be directly recovered in terms of $\mathbb{V}[v_*]$ using a first-order approximation,

$$\mathbb{V}[d_*] = \left(\frac{-\sqrt{t}}{\bar{v}_*}\right) \mathbb{V}[v_*] \left(\frac{-\sqrt{t}}{\bar{v}_*}\right)^\top. \tag{14}$$

B. Choice of Covariance Function

As presented in our previous work [14], we derived a covariance function that satisfies the screened Poisson (4) which is a linear PDE. Based on work presented in [57], it is possible to reformulate linear stochastic PDEs into GP regression models. The method relies on matching the power spectral density of the covariance kernel and the PDE's finite-dimension Markov process. Accordingly, using the so-called Whittle kernel [58], from the Matérn family, allows to recover a solution for the screened Poisson equation (4) as a GP regression model,

$$k(\mathbf{x}, \mathbf{x}') = \frac{|\mathbf{x} - \mathbf{x}'|}{2\lambda} K_\nu \left(\lambda |\mathbf{x} - \mathbf{x}'|\right), \tag{15}$$

where $\lambda = 1/\sqrt{t}$ and K_ν is the modified Bessel function of the second kind of order $\nu = 1$. From the GP regression viewpoint, λ is the inverse of the length scale hyperparameter of the covariance function. It controls the smoothness and interpolation ability of the model. In this paper, hyperparameter optimisation is not considered. Nevertheless, from (6) we show that when t gets closer to zero, corresponding to a larger λ, the proposed model will produce a more accurate EDF. However, a λ too large would lead to the loss of the GP's interpolation ability and cause numerical issues [14]. On the other hand, a small λ value, corresponding to a large length scale, will introduce inaccuracies to the distance field through the loss of details in the infered distance field. As the datasets used in this paper are indoor scenarios, we address this trade-off between interpolation and distance accuracy by setting λ as 20.

Unfortunately, the Whittle covariance kernel is not differentiable as Matérn kernel functions are $|\nu - 1|$ times differentiable. Therefore, the Whittle kernel cannot be used for gradient inference. Given that the standard Whittle covariance is a special case of the Matérn family of covariance functions (with $\nu = 1$), we proposed in [14] to use the Matérn $\nu = 3/2$ kernel function as it is the closest Matérn functions that is at least once differentiable. The Matérn family of covariance functions are given by:

$$k(\mathbf{x}, \mathbf{x}') = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}}{l} |\mathbf{x} - \mathbf{x}'|\right)^\nu K_\nu \left(\frac{\sqrt{2\nu}}{l} |\mathbf{x} - \mathbf{x}'|\right) \tag{16}$$

where l the characteristic length scale and σ^2 the signal's variance are the hyperparameters controlling the generalisation information of GP models, and ν manages the degree of smoothness. We empirically observed in [14] that the Matérn $3/2$ function with $l = \sqrt{2\nu}/\lambda$ is a good approximation of Whittle covariance.

Exploiting the accurate EDF with gradients in the space, not only around the boundary but also further away, establishes the mathematical foundation to solve the pose estimation problem. Furthermore, faithfully modelling the EDF with gradients is sufficient for a trajectory optimisation-based planner, which requires the distance and gradients to the nearest obstacles to explore complex surroundings. Moreover, the GPIS provides continuous and probabilistic representation for mapping and surface reconstruction. Our Log-GPIS unlocks the potential to provide a unified solution for sensor odometry, surface reconstruction, and safe navigation.

V. Log-GPIS-MOP Overview

Let us consider the Log-GPIS as a unified representation for simultaneous mapping, odometry and planning (MOP) with depth sensors. Log-GPIS-MOP aims to incrementally estimate the EDF, while at the same time localising the sensor, and with this information plan its path to avoid colliding into the surfaces implicitly described by the EDF. Fig. 2 shows an overview of the full approach, with data flows and notations later explained in the Sec. VI, VII, and VIII. Given the input data from any type of depth sensor, e.g. LiDAR and depth cameras, the pose estimation problem for incremental odometry is formulated as iterative alignment where the Log-GPIS incremental map is used to minimise the distance from a newly arrived point cloud to the existing surface following the gradients. The incremental mapping consists in fusing the current point cloud into the existing Log-GPIS that probabilistically represents the EDF and its gradient. After a post-processing optimisation for a batch of poses, a post-processing mapping is applied based on the structured kernel interpolation framework with derivatives (D-SKI) [59]. Finally, a path planning approach uses the Log-GPIS incremental mapping to avoid the surfaces of the mapped environment using the EDF and gradient information. Note that Log-GPIS-MOP is based on the assumption that the environment has sufficient curvature to unequivocally recover the pose of the sensor at any given sensor frame.

VI. Odometry

Let us consider a depth sensor moving in a static environment. The sensor captures a sequence of organised or unorganised point clouds as measurements of the environment. Let us assume that the measurement noise is independent and Gaussian distributed. The sensor frame at time t_i is denoted as

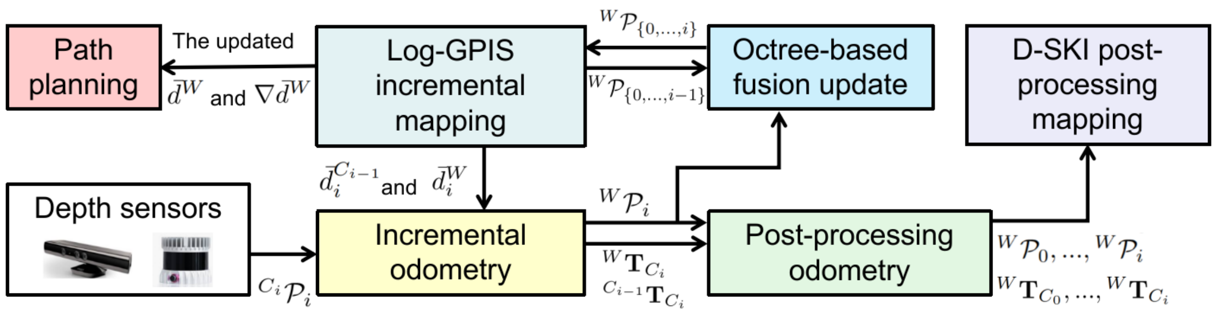

Path planning
The updated
$\bar{d}^W$ and $\nabla\bar{d}^W$
Log-GPIS incremental mapping
${}^W\mathcal{P}_{\{0,...,i\}}$
${}^W\mathcal{P}_{\{0,...,i-1\}}$
Octree-based fusion update
D-SKI post-processing mapping
$\bar{d}_i^{C_{i-1}}$ and $\bar{d}_i^W$
Depth sensors
Incremental odometry
${}^{C_i}\mathcal{P}_i$
${}^W\mathcal{P}_i$
${}^W\mathbf{T}_{C_i}$
${}^{C_{i-1}}\mathbf{T}_{C_i}$
Post-processing odometry
${}^W\mathcal{P}_0, ..., {}^W\mathcal{P}_i$
${}^W\mathbf{T}_{C_0}, ..., {}^W\mathbf{T}_{C_i}$

Fig. 2. Block Diagram of Log-GPIS-MOP framework. Depth sensors capture raw measurements as a sequence of point clouds ${}^{C_i}\mathcal{P}_i$. Incremental odometry computes the frame-to-frame poses ${}^{C_{i-1}}\mathbf{T}_{C_i}$ and frame-to-map poses ${}^W\mathbf{T}_{C_i}$ based on the local Log-GPIS representation $\bar{d}_i^{C_{i-1}}$ and the global Log-GPIS representation $\bar{d}_i^W$. Applying the estimated frame-to-map pose, the current point cloud ${}^W\mathcal{P}_i$ in the global frame then fuses with the existing points ${}^W\mathcal{P}_{\{0,...,i-1\}}$ into ${}^W\mathcal{P}_{\{0,...,i\}}$ to update the global Log-GPIS representation $\bar{d}_i^W$. The full updated representation $\bar{d}^W$ with gradients $\nabla\bar{d}^W$ is used for optimisation-based path planning. The post-processing odometry optimises a batch of poses and then the post-processing mapping uses a sequence of point clouds and optimised poses as input to create the map.

$\mathfrak{F}_{C_i}$, where $i = (0, ..., L)$. The world reference $\mathfrak{F}_W$ is given by the sensor frame $\mathfrak{F}_{C_0}$ at initial the timestamp. The pose of $\mathfrak{F}_{C_i}$ in world reference $\mathfrak{F}_W$ is given by the rotation matrix ${}^W\mathbf{R}_{C_i} \in \mathrm{SO}(3)$ and the translation vector ${}^W\mathbf{t}_{C_i}$, combined into the 4×4 homogeneous transformation matrix $\in \mathrm{SE}(3)$ as,

$$ {}^W\mathbf{T}_{C_i} = \begin{bmatrix} {}^W\mathbf{R}_{C_i} & {}^W\mathbf{p}_{C_i} \\ \mathbf{0}^\top & 1 \end{bmatrix}. \tag{17} $$

We denote the point cloud measurement of the sensor frame $\mathfrak{F}_{C_i}$ at time t_i as ${}^{C_i}\mathcal{P}_i$ and ${}^{C_i}\mathbf{p}_{i,j} \in \mathbb{R}^3$ the j-th point of ${}^{C_i}\mathcal{P}_i$ with $j = 0, ..., J_i$. This 3D point cloud ${}^{C_i}\mathcal{P}_i$ can be projected from $\mathfrak{F}_{C_i}$ to $\mathfrak{F}_W$ using ${}^W\mathbf{T}_{C_i}$,

$$ \begin{bmatrix} {}^W\mathcal{P}_i \\ \mathbf{1} \end{bmatrix} = {}^W\mathbf{T}_{C_i} \begin{bmatrix} {}^{C_i}\mathcal{P}_i \\ \mathbf{1} \end{bmatrix}. \tag{18} $$

Also, let us express ${}^W\mathbf{T}_{C_i}$ through the concatenation of ${}^W\mathbf{T}_{C_{i-1}}$ and ${}^{C_{i-1}}\mathbf{T}_{C_i}$ as:

$$ {}^W\mathbf{T}_{C_i} = {}^W\mathbf{T}_{C_{i-1}} {}^{C_{i-1}}\mathbf{T}_{C_i}, \tag{19} $$

where ${}^W\mathbf{T}_{C_{i-1}}$ is given by the odometry computation at time t_{i-1}. We aim to find the ${}^W\mathbf{T}_{C_i}$ such that the newly arrived ${}^{C_i}\mathcal{P}_i$ can be projected to the world frame.

We present first the incremental formulation that leverages directly the Log-GPIS. This incremental odometry formulation is the one used by the incremental mapping and planning approaches all tightly integrated through the Log-GPIS. Furthermore, we also present the batch optimisation odometry formulation that can be used as a post-processing step to refine the trajectory estimation. Note that the batch formulation uses the sequential poses only and not directly the Log-GPIS.

A. Incremental Formulation

In this work, we propose an iterative approach to estimate the odometry based on the Log-GPIS representation. Our odometry estimation approach follows an iterative optimisation to estimate the frame transformations. This is performed in a similar way to the Iterative Closest Point (ICP)[19] but without looking for the closest points.

Our work proposes instead, a direct query of the distance to the surface based on Log-GPIS to find the alignment efficiently and accurately without the need for the closest points search. Given a fused global representation with the information of all previous frames, the minimum distance to the surface can be obtained directly from the Log-GPIS, which saves us from doing the correspondences search. With the assumption that the difference between two consecutive frames is relatively small, the transformation is computed via a non-linear least-square optimisation involving the distances queried from the Log-GPIS effectively corresponding to point-to-plane constraints. After the transformation of the current frame is applied, the current point cloud is fused to the global representation using a fusion update method, which will be discussed in Sec. VII-A.

More formally, let us consider a local Log-GPIS representation $\bar{d}_i^{C_{i-1}}$ obtained using the previous frame only and a global Log-GPIS representation $\bar{d}_i^W$, which is the result of fusing all the previous frames into the world frame. Let us assume the initial ${}^W\mathbf{T}_{C_0}$ is identity and define $\mathcal{X}_i = \{{}^W\mathbf{T}_{C_1}, ..., {}^W\mathbf{T}_{C_i}\}$ with $i = (0, ..., L)$ as the state to be estimated. The odometry is then formulated incrementally to estimate ${}^W\mathbf{T}_{C_i}$ through a frame-to-frame alignment Fig. 3, with the possibility to compute the frame-to-map alignment as shown in Fig. 4.

The iterative alignment in both cases, frame-to-map and frame-to-frame, is performed by minimising a cost function e^i_{dis},

$$ \check{\mathcal{X}}_i = \arg\min_{\mathcal{X}_i} e^i_{\text{dis}}, \tag{20} $$

where e^i_{dis} is a scalar and corresponds to the sum of the squared distances from each points in the current point cloud ${}^{C_i}\mathcal{P}_i$ to the surface representation Log-GPIS. Frame-to-frame case uses the local surface representation and frame-to-map case uses the global surface representation. $\check{\mathcal{X}}_i$ is the sequential estimation of the current pose.

a) Frame-to-frame: This alignment is computed through the current frame i querying the distance from the representation built from the previous sensor frame C_{i-1} only (the local Log-GPIS $\bar{d}_i^{C_{i-1}}$).

Particularising (20) for this case (Fig. 3), let us write e^i_{dis} as:

$$ e^i_{\text{dis}}\left({}^{C_{i-1}}\mathbf{T}_{C_i}\right) = \sum_{j=0}^{J_i} \left\| \bar{d}_i^{C_{i-1}}\left({}^{C_{i-1}}\mathbf{T}_{C_i} {}^{C_i}\mathbf{p}_{i,j}\right) \right\|^2_{\sigma^i_{dis}}, \tag{21} $$

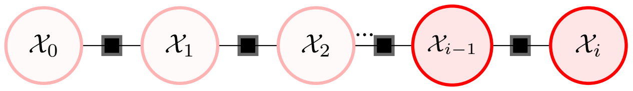


Fig. 3. Factor graph representation of the frame-to-frame odometry. $\mathcal{X}_i = \{{}^W\mathbf{T}_{C_1}, ..., {}^W\mathbf{T}_{C_i}\}$ with $i = (0, ..., L)$. Low opacity nodes are used for those nodes that are not used for current alignment. Black factors are the local Log-GPIS distance constraints.

where σ^i_{dis} is the distance variance from the Log-GPIS inference. Given the local Log-GPIS for an arbitrary testing point ${}^{C_{i-1}}\mathbf{p}_{i,j}$, the predictive mean $\bar{v}_i^{C_{i-1}}$ with derivatives are obtained from (9). Also, as per (12), $\bar{d}_i^{C_{i-1}}$ can be computed by applying the logarithm transformation to $\bar{v}_i^{C_{i-1}}$,

$$\begin{bmatrix} \bar{v}_i^{C_{i-1}} \\ \nabla \bar{v}_i^{C_{i-1}} \end{bmatrix} = (\tilde{\mathbf{k}}^{C_{i-1}})^\top (\tilde{K}^{C_{i-1}} + \sigma^2 I)^{-1} \begin{bmatrix} \mathbf{y}^{C_{i-1}} \\ \nabla \mathbf{y}^{C_{i-1}} \end{bmatrix} \tag{22}$$

$$\bar{d}_i^{C_{i-1}} = -\sqrt{t} \ln \bar{v}_i^{C_{i-1}}. \tag{23}$$

$\tilde{K}^{C_{i-1}}$ is the covariance matrix of the input points ${}^{C_{i-1}}\mathcal{P}_{i-1}$ and $\tilde{\mathbf{k}}^{C_{i-1}}$ represents the covariance vector between the input points and the testing point ${}^{C_{i-1}}\mathbf{p}_{i,j}$. $\mathbf{y}^{C_{i-1}}$ are the measurements of ${}^{C_{i-1}}\mathcal{P}_{i-1}$ of frame $\mathfrak{F}_{C_{i-1}}$ at t_{i-1} and $\nabla \mathbf{y}^{C_{i-1}}$ are the noisy surface normals computed from the pointcloud (normal computation is further explained in Sec. VII-A).

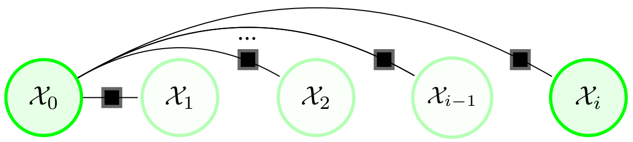


Fig. 4. Factor graph for the frame-to-map odometry. Black factors represent the global Log-GPIS distance constraints.

As in any frame-to-frame odometry, it can easily drift. The other downside of this simple alignment is that the local log-GPIS has to be created at every step and discarded afterwards. This motivates the use of a more robust frame-to-map alignment, which uses the fused global Log-GPIS.

b) Frame-to-Map: Let us consider a sequentially computed global Log-GPIS $\bar{d}_i^W$, which uses frame-to-map alignment to compute the poses as the sensor moves through the environment. The global Log-GPIS representation is the result of fusing all the previous frames into a single representation in the world coordinate frame. Particularising e^i_{dis} for frame-to-map odometry (Fig. 4), we have:

$$e^i_{\text{dis}}\left({}^W\mathbf{T}_{C_i}\right) = \sum_{j=0}^{J_i} \left\| \bar{d}_i^W \left({}^W\mathbf{T}_{C_i}{}^{C_i}\mathbf{p}_{i,j}\right) \right\|^2_{\sigma^i_{dis}}, \tag{24}$$

where the global Log-GPIS $\bar{d}_i^W$ is computed from,

$$\begin{bmatrix} \bar{v}_i^W \\ \nabla \bar{v}_i^W \end{bmatrix} = (\tilde{\mathbf{k}}^W)^\top (\tilde{K}^W + \sigma^2 I)^{-1} \begin{bmatrix} \mathbf{y}^W \\ \nabla \mathbf{y}^W \end{bmatrix} \tag{25}$$

$$\bar{d}_i^W = -\sqrt{t} \ln \bar{v}_i^W, \tag{26}$$

with $\tilde{K}^W$ the covariance matrix of all the fused point clouds from 0 to $i-1$ in world reference frame ${}^W\mathcal{P}_{\{0,...,i-1\}}$ and $\tilde{\mathbf{k}}^W$ the covariance vector between ${}^W\mathcal{P}_{\{0,...,i-1\}}$ and the testing point ${}^W\mathbf{p}_{i,j}$. $\mathbf{y}^W$ and $\nabla\mathbf{y}^W$ are the transformed implicit surface value and the normal computed from the pointcloud of ${}^W\mathcal{P}_{\{0,...,i-1\}}$.

Further, to solve the optimisation of the alignment problem, the derivative of the distance with respect to ${}^W\mathbf{T}_{C_i}$ is desired. Based on the chain rule, the Jacobian function is as follows,

$$\frac{\partial \bar{d}_i^W \left({}^W\mathbf{T}_{C_i}{}^{C_i}\mathbf{p}_{i,j}\right)}{\partial \left({}^W\mathbf{T}_{C_i}\right)} = \frac{\partial \bar{d}_i^W \left({}^W\mathbf{T}_{C_i}{}^{C_i}\mathbf{p}_{i,j}\right)}{\partial \left({}^W\mathbf{T}_{C_i}{}^{C_i}\mathbf{p}_{i,j}\right)} \frac{\partial \left({}^W\mathbf{T}_{C_i}{}^{C_i}\mathbf{p}_{i,j}\right)}{\partial \left({}^W\mathbf{T}_{C_i}\right)} \tag{27}$$

The first term is one of the valuable byproducts of the global Log-GPIS, which is the gradient of query point ${}^W\mathbf{T}_{C_i}{}^{C_i}\mathbf{p}_{i,j}$:

$$\frac{\partial \bar{d}_i^W \left({}^W\mathbf{T}_{C_i}{}^{C_i}\mathbf{p}_{i,j}\right)}{\partial \left({}^W\mathbf{T}_{C_i}{}^{C_i}\mathbf{p}_{i,j}\right)} = \nabla \bar{d}_i^W \left({}^W\mathbf{T}_{C_i}{}^{C_i}\mathbf{p}_{i,j}\right). \tag{28}$$

The second term is the derivative of ${}^W\mathbf{T}_{C_i}$ with respect to the point ${}^{C_i}\mathbf{p}_{i,j}$. It is equivalent to

$$\frac{\partial \left({}^W\mathbf{T}_{C_i}{}^{C_i}\mathbf{p}_{i,j}\right)}{\partial \left({}^W\mathbf{T}_{C_i}\right)} = \begin{bmatrix} \boldsymbol{I} & -\left({}^W\mathbf{T}_{C_i}{}^{C_i}\mathbf{p}_{i,j}\right)^\wedge \\ \mathbf{0}^T & \mathbf{0}^T \end{bmatrix}, \tag{29}$$

where $\wedge$ is the skew-symmetric operator for the non-homogeneous 3D point $\mathbf{p}$,

$$\mathbf{p}^\wedge = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix}^\wedge = \begin{bmatrix} 0 & -p_3 & p_2 \\ p_3 & 0 & -p_1 \\ -p_2 & p_1 & 0 \end{bmatrix}. \tag{30}$$

Using the querying distance and analytical Jacobian function, we implement our proposed odometry in Ceres [5].

B. Post Process Batch Optimisation

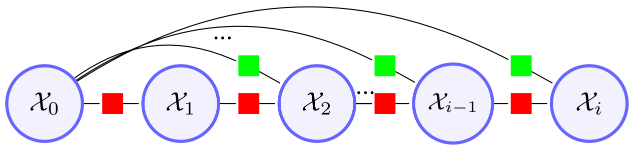


Fig. 5. Factor graph representation of the pose SLAM problem. Green factors represent the frame-to-map constraints and red factors are the frame-to-frame constraints.

Given the incremental pose estimation with frame-to-frame and frame-to-map transformations, it is possible to optimise a batch of poses through a pose graph optimisation (PGO) [1], [60], [61]. Note that the batch optimisation presented is an offline process to refine the map, and it is not used during incremental mapping and planning. As shown in Fig. 5, blue circles are the vertices representing poses. Green and red factors are the constrained edges, which represent the relations between poses. The edges are computed from the frame-to-map and frame-to-frame odometry respectively. We employ the standard PGO formulation as a Maximum Likelihood Estimation (MLE) [1], [60], [61]:

$$\hat{\mathcal{X}} = \underset{\mathcal{X}}{\operatorname{argmin}} - \log(p(\mathcal{X} \mid \check{\mathcal{X}})) = \arg\min_{\mathcal{X}} e_{pgo}, \tag{31}$$

[5]https://ceres-solver.org

where $\hat{\mathcal{X}}$ is the state of poses and, e_{pgo} is the error of the poses. Here, the PGO takes the frame-to-map constraints (in green measurements) and frame-to-frame constraints (in red measurements) to formulate the e_{pgo} [1], [61]:

$$e_{\text{pgo}}\left({}^{W}\mathbf{T}_{C_i}\right) = \sum_{i\in 0,.,L} \left\|\log\left({}^{W}\check{\mathbf{T}}_{C_i}^{-1}{}^{W}\mathbf{T}_{C_i}\right)\right\|^2_{\Sigma^i_{gpo}}, \tag{32}$$

where ${}^{W}\check{\mathbf{T}}_{C_i}^{-1}$ is the noisy estimation of ${}^{W}\mathbf{T}_{C_i}$. Σ^i_{pgo} is the covariance of poses.

VII. Mapping

We present two different pipelines, one online and one offline. The incremental mapping aims to sequentially fuse new incoming point clouds into the global Log-GPIS representation for odometry and planning. The post-processing mapping uses the output of the post-processing odometry and the raw point clouds to recover the full reconstruction for visualisation.

A. Incremental Mapping

After the pose is estimated and applied as described in Sec. VI, the current frame needs to be fused/appended with the global Log-GPIS representation for incremental mapping. To do so, we adopt the approach proposed in [12] for GPIS-SDF. This approach uses an Octree-based Gaussian Process regressor and Bayesian fusion to merge old measurements and new observations.

We briefly summarise the process here, which has been generalised for organised and unorganised point clouds. To initially remove outliers, we follow the neighbourhood filtering technique in [62] reducing excessive noise without penalising important details. Moreover, we set a minimum and maximum range for the sensor data. Points beyond the maximum range and closer than the minimum range are considered invalid. Let us then consider the current point cloud ${}^{W}\mathcal{P}_i$ and all fused global points ${}^{W}\mathcal{P}_{\{0,\ldots,i-1\}}$. The objective of incremental mapping is to have the new fused global point cloud ${}^{W}\mathcal{P}_{\{0,\ldots,i\}}$. The full point cloud ${}^{W}\mathcal{P}_{\{0,\ldots,i-1\}}$ is stored in an Octree-based structure to reduce the computational complexity, which divides the complete data into many overlapping clusters. The clusters within the same region as ${}^{W}\mathcal{P}_i$ are set as active clusters and needs to be fused with ${}^{W}\mathcal{P}_i$. In order to fuse ${}^{W}\mathcal{P}_i$ with active clusters in ${}^{W}\mathcal{P}_{\{0,\ldots,i-1\}}$, let us model a GP regressor using ${}^{W}\mathcal{P}_i$ in terms of the bearing angles α and ε of each point ${}^{W}\mathbf{p} \in {}^{W}\mathcal{P}_i$ and its inverse range for either unorganised or organised point clouds as,

$$\rho^{-1} \sim \mathcal{GP}\{0, k((\alpha,\varepsilon),(\alpha',\varepsilon'))\}. \tag{33}$$

The Ornsten-Uhlenbeck covariance [54] is used in this case as it captures more details without strong smoothing. Note that for organised point clouds the conversion from pixel coordinates to bearing angles given the camera calibration is straightforward. Then GP regressor can infer smoothly and probabilistically the reciprocal range value at any given point with values α and ε. Each active point ${}^{W}\mathbf{p}'$ in the Octree of the existing representation ${}^{W}\mathcal{P}_{\{0,\ldots,i-1\}}$ needs to be fused with ${}^{W}\mathcal{P}_i$ using the above GP regressor.

The regressor accepts the bearings of ${}^{W}\mathbf{p}'$ to infer the range value $\bar{\rho}^{-1}$ as in (33). Then, we test the occupancy function Φ for ${}^{W}\mathbf{p}'$ below,

$$\Phi = \frac{2}{1+\exp\left(-\eta\left(\rho'^{-1} - \bar{\rho}^{-1}\right)\right)}, \tag{34}$$

where ρ' is the given range of the point and $\bar{\rho}$ is the range inference. η is a slope parameter. The point is free with $\rho' > \bar{\rho}$, and occupied with $\rho' < \bar{\rho}$.

If the occupancy value is larger than a threshold, then the target point moving procedure is performed. With a negative occupancy value beyond the surface, it moves one step along the direction of the surface normal. On the contrary, with a positive value, it moves in the opposite direction of normal. Iteratively, using the regressor, the algorithm ends up with a new point ${}^{W}\mathbf{p}''$ very close to the surface. The normal of ${}^{W}\mathbf{p}''$ is computed using $\left(\Phi\left({}^{W}\mathbf{p}'' + \delta\boldsymbol{e}_v\right) - \Phi\left({}^{W}\mathbf{p}'' - \delta\boldsymbol{e}_v\right)\right)/2\delta$, where δ is a small positive value and $\boldsymbol{e}_v$ is unit vector along each normal axis.

Given ${}^{W}\mathbf{p}'$ with variance σ' (in the existing global Log-GPIS) and the inferred point ${}^{W}\mathbf{p}''$ with variance σ'' (in the current frame) both in the overlapping part, the Bayesian fusion directly updates the surface point ${}^{W}\mathbf{p}''$ with variance σ'' that implicitly will update each points in active clusters for the global Log-GPIS representation using:

$$\begin{aligned} {}^{W}\mathbf{p}'' &\leftarrow \frac{\sigma''{}^{W}\mathbf{p}' + \sigma'{}^{W}\mathbf{p}''}{\sigma'' + \sigma'} \\ (\sigma'')^{-1} &\leftarrow (\sigma'')^{-1} + (\sigma')^{-1}. \end{aligned} \tag{35}$$

B. Post Processing Mapping

Given the final estimated transformations ${}^{W}\mathbf{T}_{C_i}$ for the sequence of point clouds ${}^{C_i}\mathcal{P}_i$, which can be naively merged into one point cloud in the world reference frame, we aim to build a globally consistent continuous and probabilistic reconstruction. However, a well-known disadvantage of general GPs lies in the memory allocation and the computational complexity to invert the covariance matrix of size J the number of training points. Moreover, as pointed out above, we model the joint GPIS with gradients, which improves the accuracy of the predicted surfaces. However, considering the gradients as the input increases the size of the covariance matrix by a factor of four. This significantly worsens the computational cost of the plain GPIS.

To reduce the general computational complexity, and thereby improve scalability, in our previous work [10], we adopted the idea of the Structured Kernel Interpolation algorithm (SKI) [63] to approximate the exact kernel matrix through interpolation weights and inducing points, and extend to D-SKI when considering the derivatives with SKI [59], [64]. The input dataset is reprojected onto a grid generated by inducing points, which significantly reduces the requirement to compute GPIS. The interpolated covariance function of $k\left(\mathbf{x},\mathbf{x}'\right)$ is formulated as,

$$K \approx VK(U,U)V^T, \tag{36}$$

where U is a uniform grid of inducing points m. V is a $J\times m$ sparse matrix of weights used for interpolation, which intuitively represents the relative distances from input points to inducing points [63]. Formally, we use quintic [65] interpolation to obtain V. When considering gradients and preserving the positive definiteness of the approximate kernel, D-SKI differentiates the combined covariance matrix. The approximate covariance matrix with derivatives of $\tilde{k}(\mathbf{x},\mathbf{x}')$ is:

$$\tilde{K} \approx \begin{bmatrix} V \\ \nabla V \end{bmatrix} K(U,U) \begin{bmatrix} V \\ \nabla V \end{bmatrix}^T = \begin{bmatrix} VK(U,U)V^T & VK(U,U)(\nabla V)^T \\ (\nabla V)K(U,U)V^T & (\nabla V)K(U,U)(\nabla V)^T \end{bmatrix}, \quad (37)$$

where ∂V is the derivative of V.

C. Marching Cubes for Log-GPIS

As we mentioned in Sec. IV-A, the surface is no longer at the zero crossing and the sign information is lost. Thus, to obtain the mesh for dense reconstruction, it requires a modified version of the marching cubes algorithm [66]. Given a set of query points for Log-GPIS to infer the implicit surface values, the marching cubes compute and extracts the surface by connecting points of a constant value (iso-value) within a volume of space:

$$\mathbf{FV} = f_{iso}(\mathcal{M}, v_{iso}), \quad (38)$$

where $\mathcal{M}$ specifies a set of points for querying. v_{iso} is the iso-value at which to compute the surface, specified as a scalar. $\mathbf{FV}$ contains the computed faces and vertices of the surface. Note that since the sign is lost, instead of going through exactly zero crossing, our marching cubes sets the iso-value as a constant value (0.1m for 2D cases and 0.001m for 3D in our experiments). Then, the computed vertices in $\mathbf{FV}$ move towards the implicit surface and iteratively query the Log-GPIS until the local minimum is found (iso-values are smaller than a threshold. We set 0.0001m in our experiments).

VIII. Planning

The differentiable nature of Log-GPIS permits straightforward integration with existing optimisation-based planners. In this section, we present how Log-GPIS can be used together with the covariant Hamiltonian optimisation-based motion planner (CHOMP) [7] for obstacle avoidance planning.

Let $\mathbf{x}^r : t \mapsto \mathbf{x}^r(t)$ be the robot's trajectory. For simplicity, we consider the robot's position in the workspace over time, excluding orientation, so that $\mathbf{x}^r(t) \in \mathbb{R}^D$. However, the framework can be easily extended to cases with orientation or with multiple joints.

CHOMP minimises an objective *functional* defined over a space of trajectories. We consider the conventional objective function as introduced in [7] for smooth obstacle avoidance:

$$\mathcal{C}[\mathbf{x}^r] \equiv \int_0^T \frac{1}{2}||\dot{\mathbf{x}}^r(t)||^2 + \lambda c(\mathbf{x}^r(t))||\dot{\mathbf{x}}^r(t)||dt. \quad (39)$$

Here, the first term encourages smoothness by regularising the velocity. The second term is responsible for obstacle avoidance, which is enforced by the collision penalty term $c(\mathbf{x}^r(t))$ set as:

$$c(\mathbf{x}^r(t)) = \begin{cases} -d(\mathbf{x}^r(t)) + \frac{1}{2}\epsilon, & \text{if } d(\mathbf{x}^r(t)) < 0 \\ \frac{1}{2\epsilon}(d(\mathbf{x}^r(t)) - \epsilon)^2, & \text{if } 0 < d(\mathbf{x}^r(t)) \leq \epsilon \\ 0, & \text{otherwise.} \end{cases} \quad (40)$$

CHOMP incrementally minimises the objective functional $\mathcal{C}[\mathbf{x}^r]$ using its functional gradient[6]. With the objective set as (39), the functional gradient is given by:

$$\nabla\mathcal{C}[\mathbf{x}^r] = -\ddot{\mathbf{x}}^r + ||\dot{\mathbf{x}}^r|| \left(\left(I - \dot{\mathbf{x}}^r\dot{\mathbf{x}}^{rT}\right)\nabla c(\mathbf{x}^r) - c(\mathbf{x}^r)\kappa\right). \quad (41)$$

Here, $\kappa = ||\dot{\mathbf{x}}^r||^{-2}\left(I - \dot{\mathbf{x}}^r\dot{\mathbf{x}}^{rT}\right)\ddot{\mathbf{x}}^r$ is the curvature vector. The gradient of the collision penalty term $c(\mathbf{x}^r)$ is given by

$$\nabla c(\mathbf{x}^r) = \begin{cases} -\nabla d(\mathbf{x}^r), & \text{if } d(\mathbf{x}^r) < 0 \\ (d(\mathbf{x}^r) - \epsilon)\nabla d(\mathbf{x}^r) & \text{if } 0 < d(\mathbf{x}^r) \leq \epsilon \\ 0, & \text{otherwise.} \end{cases} \quad (42)$$

The merit of our framework is that we can use the Log-GPIS gradient inference equation (13) to efficiently and accurately compute the gradient $\nabla d(\mathbf{x}^r)$, and, in turn, (41), (42). This is because our formulation produces the gradients analytically, whereas conventional approaches rely on discrete grid-based approximation [7], [35].

IX. Experimental Results

In this section, we illustrate the performance of the proposed framework both qualitatively and quantitatively. The framework is implemented in C++ and Matlab. All experiments were run on an 8-core i7 CPU at 2.5GHz.

A. Effects of Log Transformation

In [14], we showed that our Log-GPIS has the advantage of both mapping and distance accurate estimation. In this section, we examine the effects of the log transformation on mapping, odometry and planning. We first illustrate the nominal behaviour of the full Log-GPIS-MOP framework. To do so, we first use a 20×16m simulated 2D LiDAR dataset from [12], [28], the second simulated dataset in [67], and the real-world Intel Research Lab dataset [2]. To emulate online planning with incremental mapping, we generate the plan from the robot's current pose to its future pose in the dataset, 50 time-steps ahead. Although the robot does not strictly follow the generated plan, this serves as a useful proxy for performance in online scenarios.

The result for the first simulated dataset is shown in Fig. 6, with the travelled trajectory in red, the planned trajectory in white, and the underlying colourmap representing the EDF built so far. Most notably, in Fig. 6c), it can be seen that the white planned trajectory avoids the obstacles even when they are not yet seen. The same behaviour is observed in the more challenging scenario of the Intel Research Lab dataset [2] illustrated in Fig. 1, where the white planned trajectories are smooth and stay a fixed distance away from the

[6]We focus only on gradient computation and omit the exact update process. Interested readers are referred to [7].

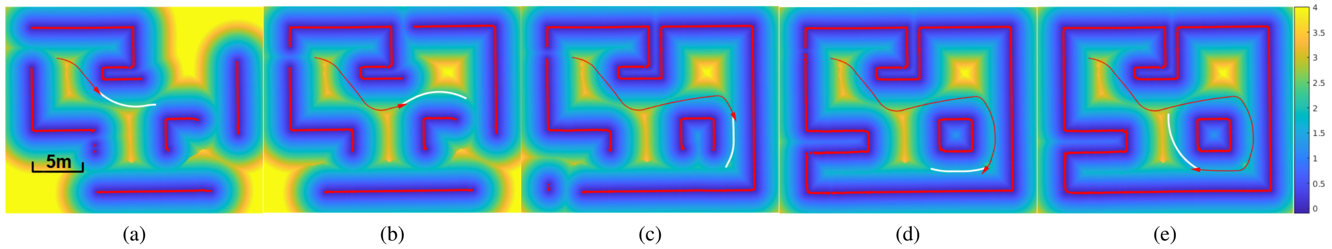


Fig. 6. The Log-GPIS-MOP on a simulated 2D dataset. The reconstructed surfaces are red lines. The colourmap represents the distance field varying from -4 to 4 meters. Red line with an arrow is the estimated trajectory where the robot travelled. The white line is the suggested path computed by the path planner using the global Log-GPIS.

obstacles as intended. This is because Log-GPIS extrapolates well to areas with missing data. Moreover, in both cases, Log-GPIS-MOP produces accurate odometry and EDF, as was observed in Sec. IX-B. We attribute such extrapolation to the log transformation, as it implicitly approximates the Eikonal equation across the entire space.

We now interrogate this hypothesis through comparison against GPIS-SDF [12] on the second simulated dataset. For fairness, we implemented the same odometry and planning algorithms using GPIS-SDF. In particular, we use the same weights for trajectory smoothness and collision avoidance when planning the trajectories.

The results are shown in Fig. 7. We observed that odometry based on GPIS-SDF fails with a moderately large motion because GPIS-SDF produces increasingly inaccurate EDF predictions further away from previous measurements, unlike Log-GPIS. Thus, we present a portion of the results before GPIS-SDF fails, in order to compare the planned trajectories.

The planned trajectories are shown in white in Fig. 7. In Figs. 7(a-b), It can be seen that trajectory planning using GPIS-SDF results in nearly a straight line, with a near miss in Fig. 7b. On the other hand, with the same weighting for collision avoidance, Log-GPIS-MOP produces a plan strongly biased towards the medial axis of the environment (i.e. equidistant from obstacles). These observations illustrate that the log transformation in Log-GPIS-MOP represents a critical improvement in prediction accuracy that is necessary for practical use in odometry and planning.

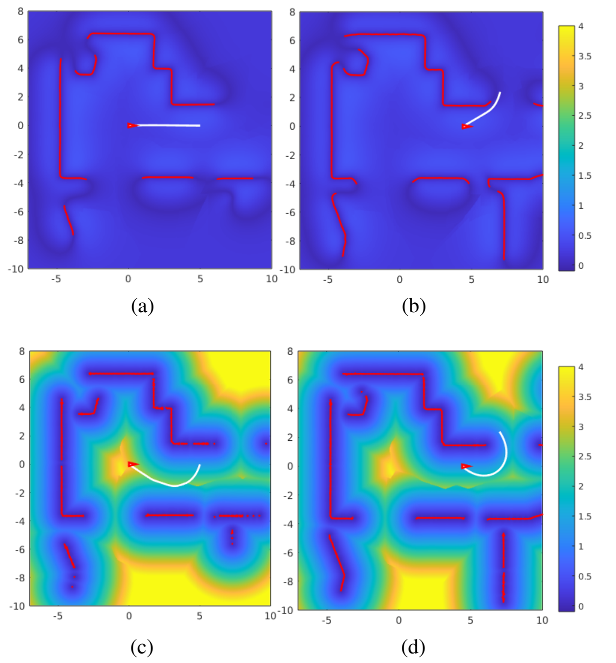


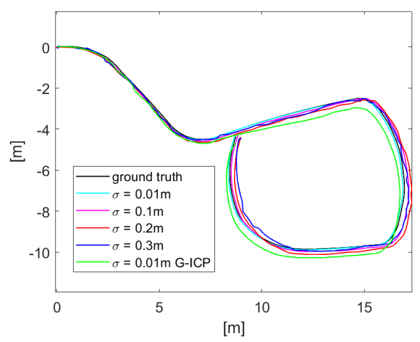
Fig. 7. Mapping and planning results of GPIS-SDF (a-b) and Log-GPIS-MOP (c-d). Red arrows show the robot's current pose. With the same collision avoidance weight, the planned trajectories (white) from Log-GPIS-MOP in c-d) are strongly biased towards the medial axis, whereas that of GPIS-SDF in a-b) are nearly straight lines, with a near-hit in b).

B. Robustness to Noise

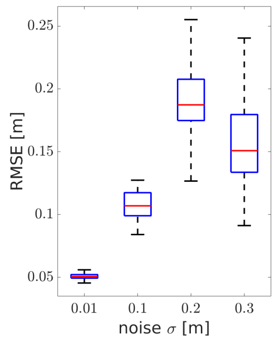
We first illustrate the robustness of the proposed Log-GPIS-MOP framework against noise. To do so, we use the same simulated 2D LiDAR dataset [12], [28] in Sec. IX-A, and add varying magnitudes of Gaussian noise from $0.01m$ to $0.3m$ to the sensor measurements. Note that this dataset is captured by a Turtlebot with a Hokuyo range sensor in Gazebo. The original standard deviation of the measurement noise is $\sigma = 0.01m$ [12]. In this simulation, we only use the sensor measurements, the sensor poses from the sensor frame at each timestamp to the world frame are estimated with our framework.

The result for odometry is shown in Fig. 8a. It can be seen that, with a realistic noise ($\sigma = 0.01m$, cyan), the result is almost identical to the ground truth trajectory (black). Even with a significantly large noise of $\sigma = 0.3m$ (blue), the estimated trajectory robustly has acceptable drift. For comparison, we plot the trajectory of Generalised-ICP (G-ICP) [68] in green. The odometry of G-ICP is frame-to-map and the sensor measurements are with Gaussian noise of $\sigma = 0.01m$. As we can see, the green line drifts quickly even with the smallest noise. Figs. 8b and 8c illustrate the root mean squared error (RMSE) over 50 Monte Carlo runs in translation and rotation respectively. It can be seen that the overall error of Log-GPIS odometry remains small, demonstrating robustness against noise. In particular, even with a large sensor noise of $\sigma = 0.3m$, the maximum translation RMSE is less than 0.25m over a 30m length trajectory.

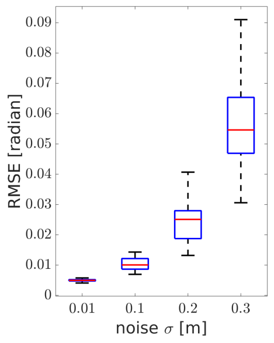
In Fig. 9, we compare the reconstructed EDF (Fig. 9b) against the ground truth (Fig. 9a). We visualise the EDF in 3D, with the z-axis being the distance field value. The colourmap varies from blue to yellow, corresponding to the distance field value from 0 to 4 meters. It can be seen that the



(a) Trajectory comparison with different noise



(b) Translation RMSE



(c) Rotation RMSE

Fig. 8. Comparison of ground truth and estimated trajectories using Log-GPIS with varying levels of simulated measurement noise. The estimated trajectories in a) closely resemble the ground truth for all noise magnitudes. The translational and rotational RMSE in b) and c) exhibits acceptable increase with noise.

EDF reconstructed using our method is close to identical to the ground truth, sharing the same peaks and valleys.

The RMSE plot in Fig. 9c further shows that the reconstructed EDF is accurate across the entire space even with increasing noise values, with a median RMSE is 0.07629m given reasonable sensor noise of 0.01m. In particular, a linear trend is observed between the noise and the distance field RMSE. This illustrates that the Log-GPIS technique can accurately predict the EDF far away from measurements even though the measurements are only available near the surface.

C. Comparison to SLAM Frameworks

1) 2D SLAM: We compare the performance of Log-GPIS MOP for 2D SLAM against Cartographer [27], a popular open-source 2D SLAM framework. Cartographer produces an occupancy grid as opposed to an EDF produced by our framework. We thus qualitatively examine the similarity of the produced maps. Nonetheless, we quantitatively compare the odometry performance. In doing so, Cartographer has the advantage that it additionally uses IMU readings for short-term odometry, as well as loop closure detection and pose-graph optimisation for long-term drift compensation. Meanwhile, our incremental Log-GPIS odometry and mapping approach solely uses LiDAR measurements. IMU fusion and loop closure remain avenues for future work.

We use two real-world datasets for this comparison, namely the Deutsches Museum [27] and the Intel Research Lab [2] datasets. The Deutsches Museum dataset was collected using a 2D LiDAR backpack. The Intel Research Lab dataset also consists of 2D LiDAR measurements (13631 frames) collected from a robot. The dataset comprises multiple loops, with two loops in the corridor followed by detailed scans of each room. Since the Deutsches Museum dataset was collected in a relatively large real-world indoor environment, there is no ground truth trajectory. On the other hand, we adopt the evaluation method suggested in [69], which computes relative poses between frames for the Intel Research Lab dataset, which we use to derive the ground truth poses.

The comparison using the Deutsches Museum dataset is shown in Fig. 10. Due to the lack of ground truth poses, we qualitatively compare the trajectory from Log-GPIS-MOP against that from Cartographer [27] as shown in Fig. 10a. Even without explicit loop closure or IMU preintegration, our incremental Log-GPIS trajectory (red line) produces a close to identical result to Cartographer (blue line). Similarly, we compare the mapping results in Figs. 10c and 10b. In Fig. 10c, the colourmap shows the distance field values from 0 (blue) to 4 (yellow) meters. The reconstructed surface (i.e. minimum absolute distance) is shown in white. We truncated the colour map at 4 meters for better visualisation of the building structures, although Log-GPIS remains accurate further away. It can be seen that the reconstructed surface closely aligns with the occupancy map from Cartographer shown in Fig. 10b.

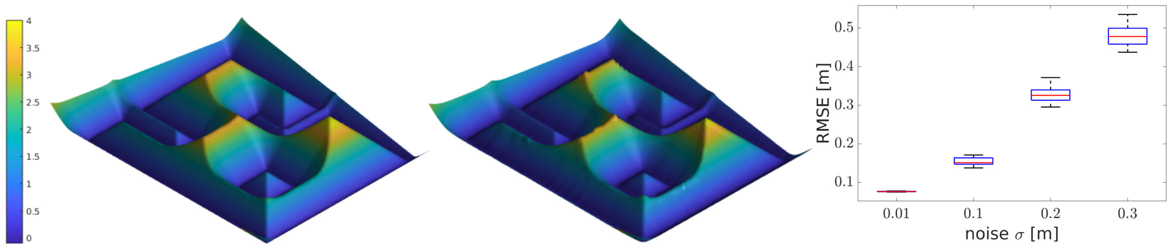
TABLE I
QUANTITATIVE COMPARISON OF ERRORS ON INTEL LAB DATASET

	Absolute translation$[m]$	*Absolute rotation*$[deg]$
Scan matching	0.220±0.296	1.7±4.8
Log-GPIS	0.0336±0.0253	1.4904±4.5507
Graph Mapping	0.031±0.026	1.3±4.7
Cartographer	0.0229±0.0239	0.453±1.335

We conduct a more quantitative analysis using the Intel Research Lab dataset [2]. We present a comparison against the Scan Matching [70] (SM), Graph Mapping [71] (GM) in addition to Cartographer, using the values reported in [27].Table. I presents the absolute translational and rotational errors with standard deviation. Overall, Cartographer exhibits the lowest error owing to loop closure detection. Our framework outperforms the SM approach and demonstrates similar results to the GM and Cartographer approaches. The mapping result in Fig. 11 shows that the framework performs as expected, providing an accurate reconstruction of the wall surface, rooms and corridors.

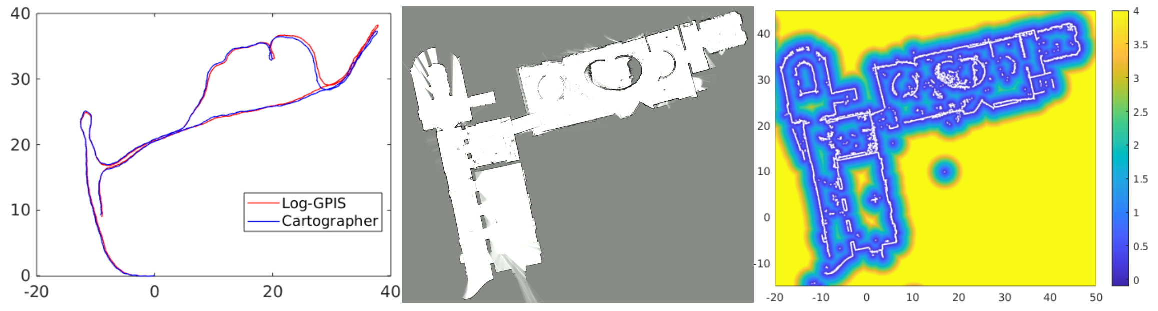
2) 3D SLAM: To evaluate the performance of Log-GPIS-MOP in the 3D setting, we compare it against ORB-SLAM2 [3], a state-of-the-art feature-based SLAM framework. ORB-SLAM2 uses sparse keypoint features from the RGB images for odometry, whereas Log-GPIS-MOP does not use the RGB component. As ORB-SLAM2 does not produce a dense map, we only compare the odometry performance. The quality of the estimated camera trajectory is evaluated using the tool from [72].

We use a real-world RGB-D dataset called the Freiburg3 Teddy from TUM [72], illustrated in Figs. 12a and 12b. The dataset consists of RGB-D images collected using a



(a) Ground Truth (b) Log-GPIS ($\sigma = 0.01m$) (c) RMSE

Fig. 9. Comparison of ground truth and estimated EDF with varying levels of noise. z-axes in b) and c) correspond to the distance value. The estimated EDF with a realistic noise $\sigma = 0.01m$ in c) is accurate and closely resembles the ground truth in b). The RMSE in d) exhibits a linear increase with noise.



(a) Estimated Trajectory (b) Cartographer (c) Log-GPIS

Fig. 10. a) Cartographer map using 2D LiDAR dataset. The resolution of the occupancy grid is 0.05 meters. The odometry and global optimisation are enabled. The rest of the parameters are set as default. b) As a comparison, the mapping result of our proposed method is shown. We can clearly see the wall and hallways. c) The surface modelling with distance field is demonstrated on the Intel dataset. The local minimum of iso-surfaces is in white colour.

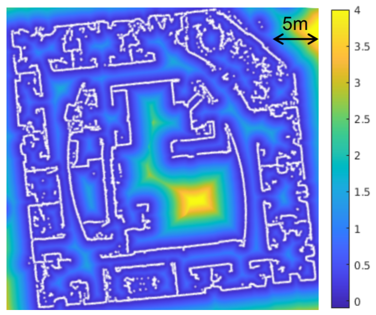


Fig. 11. Reconstructed Log-GPIS map on the Intel Research Lab dataset.

TABLE II
QUANTITATIVE COMPARISON OF ERRORS WITH ORB-SLAM2 ON A SEGMENT OF THE TEDDY BEAR DATASET

	ORB-SLAM2	Log-GPIS
Translational error RMSE [m]	0.020317	**0.017197**
Translational error mean [m]	0.017291	**0.014583**
Translational error median [m]	0.015385	**0.013027**
Translational error std [m]	0.010668	**0.009114**
Translational error max [m]	0.050399	**0.044593**
Rotational error RMSE [deg]	**1.225683**	1.452126
Rotational error mean [deg]	**1.096656**	1.289137
Rotational error median [deg]	**0.018657**	0.021269
Rotational error std [deg]	**0.547399**	0.668427
Rotational error max [deg]	**2.582371**	3.734151

Kinect sensor, along a ground-truth trajectory recorded using a motion-capture system with eight tracking cameras. The image resolution is 640×480 at a frequency of 30 Hz. We use the first 100 frames for evaluation.

The result is shown in Table. II. As can be seen, Log-GPIS-MOP outperforms ORB-SLAM in translational error, although ORB-SLAM2 shows lower rotational error. In other words, Log-GPIS-MOP exhibits comparable performance to ORB-SLAM2, even though it does not use RGB features for alignment as ORB-SLAM2 does.

D. Comparison to Distance Field Mapping Frameworks

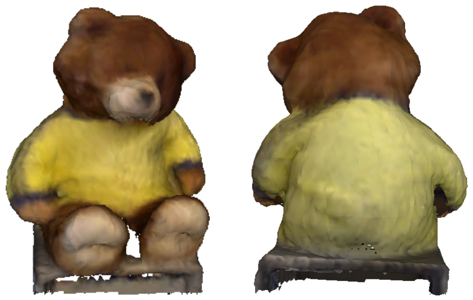
We compare the performance of Log-GPIS-MOP in surface reconstruction against Voxblox [34], a state-of-the-art distance field mapping and surface reconstruction framework. Since Voxblox is a mapping-only framework that requires external odometry, we allow Voxblox to use the *ground truth* trajectory from the dataset. Given the point cloud data and ground truth trajectory, Voxblox generates similar outputs as ours including the surface mesh, and the EDF. This is achieved by building a TSDF first, followed by wavefront propagation from some initial voxels for a certain distance to compute the EDF values. A dense mesh is reconstructed from the TSDF. The resolution and TSDF are set at 0.01m and 0.03m respectively. For a fair comparison, Log-GPIS-MOP uses a grid with the same resolution as testing points. The rest of the parameters of Voxblox are set as default.

We use two datasets, the Freiburg3 Teddy [72] used in Sec. IX-C and the Cow and Lady dataset [7]. For this comparison, we use the first 600 frames of the dataset, which corresponds to the first loop around the teddy bear.

The Cow and Lady dataset includes fibreglass models of a large cow and a lady standing side by side in a room, as

[7]https://projects.asl.ethz.ch/datasets/doku.php?id=iros2017

(a) Image of Teddy's front (b) Image of Teddy's back


(c) Voxblox + GT trajectory (front) (d) Voxblox + GT trajectory (back)

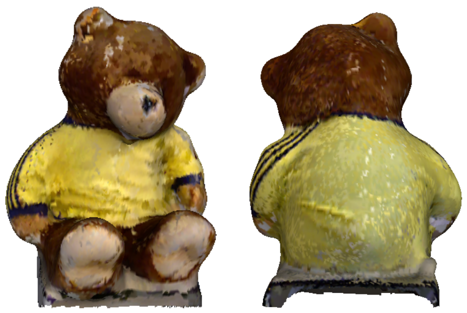

(e) Log-GPIS-MOP (front) (f) Log-GPIS-MOP (back)

Fig. 12. Qualitative evaluation on Freiburg3 Teddy dataset. a) and b) show the raw images roughly at the same angle as the reconstructed mesh. Voxblox uses the ground truth poses for the reconstruction. In contrast, our method has no prior for the localisation.

illustrated in Fig. 14a. The dataset consists of RGB-D point clouds collected using a Kinect RGB-D camera, with camera trajectory captured using a Vicon motion system. We use the first 450 frames of the dataset. This dataset is much more challenging than the Freiburg3 Teddy dataset, as the camera motion is relatively fast, with large changes in orientation. Furthermore, the Vicon trajectory is occasionally misaligned. We use Log-GPIS-MOP to refine such misalignment by using the Vicon trajectory as a prior for odometry.

The results for the Freiburg3 Teddy dataset are shown in Fig. 12. Figs. 12 c-d) and e-f) show the reconstructed map using Voxblox and Log-GPIS-MOP respectively. Most notably, it can be seen that there is a gap in the Voxblox result (Figs. 12 c-d)) between the hind paws due to missing data, whereas no such gap exists in the Log-GPIS-MOP result (Figs. 12 e-f)). This is because Log-GPIS-MOP naturally allows dealing with incomplete and sparse data, as GPIS allows extrapolation at unseen points.

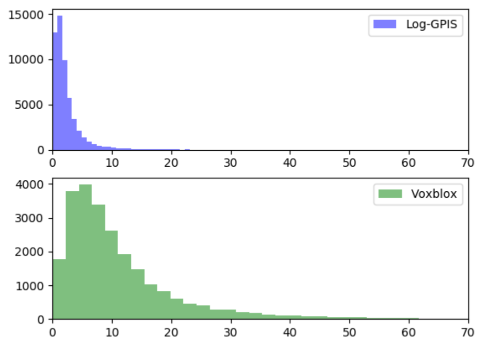


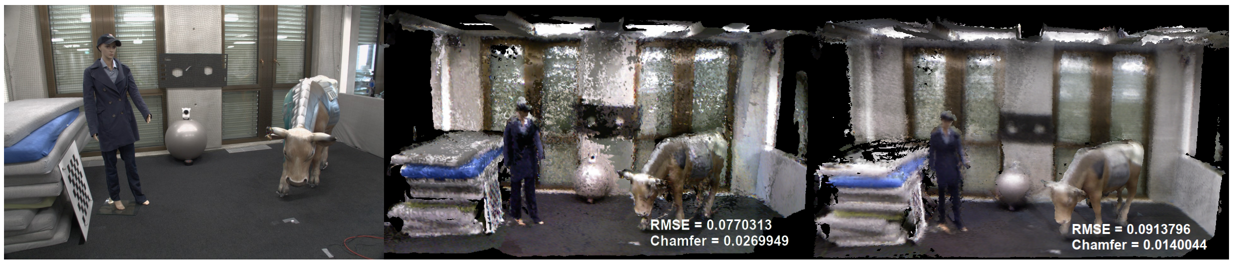
Fig. 13. A histogram of the angle distribution of the normals in degrees. The top figure is Log-GPIS and the bottom figure is Voxblox.

Qualitatively, our reconstruction is closer to the image than Voxblox, especially the head and hind paws. Some colour differences are apparent due to light source variations. Voxblox fuses the colour of each frame. In contrast, our method obtains the colour directly from the global merged point cloud, which clearly produces the three black stripes on the sleeve of the yellow shirt.

It is challenging to quantitatively evaluate the surface reconstruction performance since no ground truth is available for the Teddy bear's geometry. Instead, we examine the surface smoothness by computing the difference between the estimated normal vectors and the smoothed normal over the ten nearest neighbours. Fig. 13 shows that Log-GPIS-MOP produces a smoother surface since GPIS has the ability to filter out noisy measurements.

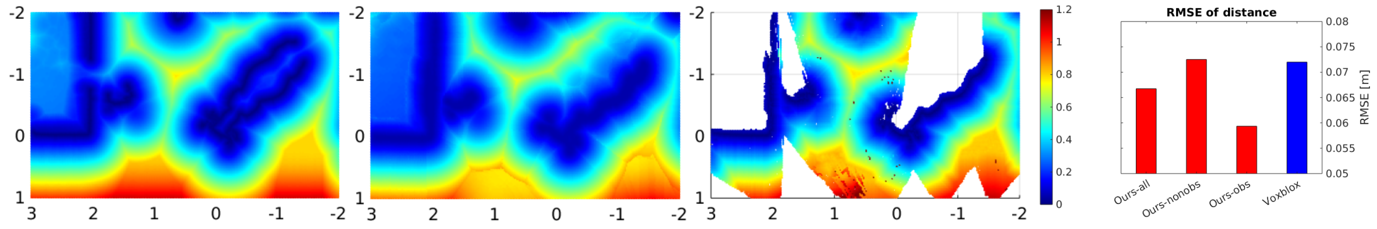
The reconstruction results on the Cow and Lady dataset using Log-GPIS-MOP and Voxblox are shown in Figs. 14. It can be seen that there is a gap at the top of the mattresses on the left-hand side in the Voxblox result due to missing data, whereas Log-GPIS-MOP does not have the same gap. This is because Log-GPIS-MOP has the advantage of extrapolating the gaps in data, as we saw in the Freiburg3 Teddy dataset. Fig. 14a shows the raw RGB image of the scene. Log-GPIS and Voxblox both produce similar reconstructed meshes shown in Fig. 14b and Fig. 14c respectively. To compare the reconstructed surfaces quantitatively, we use the RMSE and Chamfer distance metrics against Voxblox. As demonstrated in Fig. 14b, our surface quality has comparable results as Voxblox in Fig. 14c.

To further examine the behaviour of the two frameworks, we compare 2D slices of the ground-truth and estimated distance fields in Fig. 15. We choose a horizontal slice 0.8m above the ground. As shown in Fig. 15c, Voxblox only computes the EDF within the sensor's field of view, whereas Log-GPIS-MOP naturally predicts the EDF value at all points as can be seen in Fig. 15b. For a quantitative evaluation, we compare against a ground-truth distance field computed using the global point cloud from a Leica scanner. We compute the RMSE for Voxblox only within the sensor field of view, as Voxblox does not compute the EDF outside this region. Fig. 15d shows that our RMSE for the full region (Ours-all) and our prediction in observed regions (Ours-obs) clearly outperform Voxblox. The RMSE of the unobserved area (Ours-nonobs) has a similar



(a) Raw image of the scene (b) Log-GPIS mesh (c) Voxblox mesh

Fig. 14. 3D odometry and mapping evaluation on the cow and lady dataset. a) shows the raw image of the scene. b) and c) show the reconstructed meshes of our framework and Voxblox respectively. Note that Voxblox fuses the colour for the final reconstruction. In contrast, we directly take the colour from the global merged point cloud. b) and c) have the RMSE and Chamfer distance values on the right bottom as the comparison of reconstruction quality.



(a) Ground truth distance (b) Log-GPIS distance (c) Voxblox distance (d)

Fig. 15. A horizontal slice ($5m \times 3m$) shows the distance accuracy given Log-GPIS odometry and mapping on the cow and lady dataset. The slice is 0.8m above the ground. a), b), and c) show from the top view with scales in meters. For the distance RMSE in d), note that Voxblox estimates the distance for the observed area only. Our method is able to predict not only in the observed area (Ours-obs) but also extrapolate the distance field to un-observed regions (Ours-nonobs). The RMSE shows that our method outperforms Voxblox, in the observed regions, which is the one that is fair to compare with Voxblox. It also shows that the prediction of the unobserved area from our method has similar errors than Voxblox in observed areas.

performance to Voxblox in the observed regions.

Regarding the computation time, the full Log-GPIS-MOP, using for example the simulated dataset [67] (angular range from 0° to 360° with 1° resolution) consumes a median computational time of 3.63s. Individually, the frame-to-map odometry consumes 1.98s. Incremental mapping takes 0.03s and path planning uses 2.29s. The final map reconstruction including post-processing takes an additional 29.12s with 190376 querying points. For the 3D odometry and mapping on the cow and lady dataset, the median computational time for each frame is 20.38s, individually 16.59s for the frame-to-map odometry, and 3.79s for incremental mapping. We use 13517926 querying points to perform marching cube. The final dense reconstruction takes an additional 26.94 minutes. When the framework runs incrementally, the odometry is the most computationally expensive due to each frame requiring inverting the covariance matrix of the size of the training points. In the future, we will investigate the use of inducing points to make it more efficient.

To summarise, Log-GPIS-MOP has been evaluated qualitatively and quantitatively on both simulated datasets [12], [28] and public real-word datasets [2], [27], [72], [34]. Based on Log-GPIS representation, Log-GPIS-MOP provides competitive results against state-of-the-art frameworks in odometry, surface reconstruction, and obstacle avoidance. One limitation of our proposed odometry is the difficulty of finding the correct alignment in scenes lacking curvatures. This is because the distance error is the same along flat surfaces with no features. Another limitation that is present in most scan matching-based approaches is that the difference between the consecutive observations has to be relatively small in order to make sure there is sufficient overlap between consecutive measurements. Time complexity is also an issue in particular for odometry estimation that we are aiming to address in future work.

X. Conclusion

We proposed Log-GPIS-MOP, a unified probabilistic framework for mapping, odometry and planning based on Log-GPIS. The main ingredient is the Log-GPIS representation, which allows accurate prediction of the EDF and its gradients. By exploiting the global and local Log-GPIS, we presented a sequential odometry formulation for the incremental mapping and planning approaches, and a batch optimisation as post-processing to refine a sequence of poses. For Log-GPIS mapping, two different pipelines were proposed: the incremental mapping that appends the Log-GPIS with incoming point clouds and the post-processing mapping, which simply uses the output of the odometry and the raw point clouds altogether to recover a global GPIS. Concurrently, a path planning approach uses the reconstructed map to compute an optimal collision-free trajectory in the environment. Extensive analysis has been conducted to evaluate the proposed method on simulated and real datasets in both 2D and 3D against state-of-the-art frameworks. Our experiments showed that Log-GPIS-MOP exhibits comparable results to the state-of-the-art frameworks for localisation, surface mapping and obstacle avoidance, even without using IMU data or loop closure detection. Future work lies in loop closure detection to develop a full SLAM solution that will improve long-term robustness in large-scale environments. Combining the dynamic module from [55] with the proposed Log-GPIS-MOP is an interesting and promising avenue for future work that can expand the applications to more complicated scenarios. The predictive variance can be used in interesting planning problems such as information gathering [73] or planning under uncertainty [74]. Further, we would like to recover the sign of EDF and develop an efficient implementation that will allow online operation.

References

[1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. Leonard, "Past, present, and future of simultaneous localization and mapping: Towards the robust-perception age," *Trans. on Rob.*, p. 1309–1332, 2016.

[2] D. Haehnel, *Cyrill Stachniss—Robotics Datasets*, Intel Lab, 2019. [Online]. Available: http://www2.informatik.uni-freiburg.de/~stachnis/datasets/

[3] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: an open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Transactions on Robotics (T-RO)*, vol. 33, no. 5, pp. 1255–1262, 2017.

[4] J. Thoma, D. P. Paudel, A. Chhatkuli, T. Probst, and L. V. Gool, "Mapping, localization and path planning for image-based navigation using visual features and map," in *Proc. of CVPR*, 2019, pp. 7383–7391.

[5] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, 2018.

[6] 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 *Proc. of the ACM symposium*, 2011, pp. 559–568.

[7] M. Zucker, N. Ratliff, A. D. Dragan, M. Pivtoraiko, M. Klingensmith, C. M. Dellin, J. A. Bagnell, and S. S. Srinivasa, "Chomp: Covariant hamiltonian optimization for motion planning," *Int. J. of Rob. Res.(IJRR)*, pp. 1164–1193, 2013.

[8] V. Reijgwart, A. Millane, H. Oleynikova, R. Siegwart, C. Cadena, and J. Nieto, "Voxgraph: Globally consistent, volumetric mapping using signed distance function submaps," *IEEE Robotics and Automation Letters*, 2020.

[9] O. Williams and A. Fitzgibbon, "Gaussian process implicit surfaces," 2007.

[10] L. Wu, R. Falque, V. Perez-Puchalt, L. Liu, N. Pietroni, and T. Vidal-Calleja, "Skeleton-based conditionally independent gaussian process implicit surfaces for fusion in sparse to dense 3d reconstruction," *IEEE Robotics and Automation Letters*, no. 2, pp. 1532–1539, 2020.

[11] W. Martens, Y. Poffet, P. R. Soria, R. Fitch, and S. Sukkarieh, "Geometric priors for gaussian process implicit surfaces," *IEEE Robotics and Automation Letters (RA-L)*, pp. 373–380, 2017.

[12] B. Lee, C. Zhang, Z. Huang, and D. D. Lee, "Online continuous mapping using gaussian process implicit surfaces," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019.

[13] J. A. Stork and T. Stoyanov, "Ensemble of sparse gaussian process experts for implicit surface mapping with streaming data," *2020 IEEE International Conference on Robotics and Automation(ICRA)*, 2020.

[14] L. Wu, K. M. B. Lee, L. Liu, and T. Vidal-Calleja, "Faithful euclidean distance field from log-gaussian process implicit surfaces," *IEEE Robotics and Automation Letters*, pp. 2461–2468, 2021.

[15] K. Crane, C. Weischedel, and M. Wardetzky, "Geodesics in heat," *ACM Transactions on Graphics*, 2012.

[16] T. Schneider, M. T. Dymczyk, M. Fehr, K. Egger, S. Lynen, I. Gilitschenski, and R. Siegwart, "maplab: An open framework for research in visual-inertial mapping and localization," *IEEE Robotics and Automation Letters*, 2018.

[17] S. Fuhrmann, F. Langguth, and M. Goesele, "Mve: A multi-view reconstruction environment," in *Eurographics Workshop on Graphics and Cultural Heritage (GCH)*, 2014, pp. 11–18.

[18] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proc. of the Conference on Computer Graphics and Interactive Techniques*, 1996, p. 303–312.

[19] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor fusion IV: control paradigms and data structures*. International Society for Optics and Photonics, 1992, pp. 586–606.

[20] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "Rgb-d mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments," *Int. J. of Rob. Res.(IJRR)*, pp. 647–663, 2012.

[21] T. Whelan, S. Leutenegger, R. F. Salas-Moreno, B. Glocker, and A. J. Davison, "Elasticfusion: Dense slam without a pose graph," in *Robotics: Science and Systems*, 2015.

[22] E. Zobeidi, A. Koppel, and N. Atanasov, "Dense incremental metric-semantic mapping via sparse gaussian process regression," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 6180–6187.

[23] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: an open-source library for real-time metric-semantic localization and mapping," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 1689–1696.

[24] V. Vasilopoulos, G. Pavlakos, S. L. Bowman, J. D. Caporale, K. Daniilidis, G. J. Pappas, and D. E. Koditschek, "Reactive semantic planning in unexplored semantic environments using deep perceptual feedback," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4455–4462, 2020.

[25] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989.

[26] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: an efficient probabilistic 3d mapping framework based on octrees," *Autonomous Robots*, pp. 189–206, 2013.

[27] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2d lidar slam," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1271–1278.

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

[29] S. Kim and J. Kim, "Building occupancy maps with a mixture of gaussian processes," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 4756–4761.

[30] E. G. Tsardoulias, A. Iliakopoulou, A. Kargakos, and L. Petrou, "A review of global path planning methods for occupancy grid maps regardless of obstacle density," *Int. J. of Rob. Res.(IJRR)*, 2016.

[31] S. Kim and J. Kim, "Occupancy mapping and surface reconstruction using local Gaussian Processes with Kinect Sensors," in *IEEE Transactions on Cybernetics*, 2013, pp. 1335–1346.

[32] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel, "Motion planning with sequential convex optimization and convex collision checking," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1251–1270, 2014.

[33] H. Oleynikova, A. Millane, Z. Taylor, E. Galceran, J. Nieto, and R. Siegwart, "Signed distance fields: A natural representation for both mapping and planning," in *RSS 2016 Workshop: Geometry and Beyond-Representations, Physics, and Scene Understanding for Robotics*, 2016.

[34] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, "Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 1366–1373.

[35] H. Oleynikova, Z. Taylor, R. Siegwart, and J. Nieto, "Safe local exploration for replanning in cluttered unknown environments for micro-aerial vehicles," *IEEE Robotics and Automation Letters*, 2018.

[36] L. Han, F. Gao, B. Zhou, and S. Shen, "Fiesta: Fast incremental euclidean distance fields for online motion planning of aerial robots," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 4423–4430.

[37] J. Ortiz, A. Clegg, J. Dong, E. Sucar, D. Novotny, M. Zollhoefer, and M. Mukadam, "isdf: Real-time neural signed distance fields for robot perception," *arXiv preprint arXiv:2204.02296*, 2022.

[38] M. Bloesch, M. Burri, S. Omari, M. Hutter, and R. Siegwart, "Iterated extended kalman filter based visual-inertial odometry using direct photometric feedback," *Int. J. of Rob. Res.(IJRR)*, pp. 1053–1072, 2017.

[39] A. Gropp, L. Yariv, N. Haim, M. Atzmon, and Y. Lipman, "Implicit geometric regularization for learning shapes," *arXiv preprint arXiv:2002.10099*, 2020.

[40] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 165–174.

[41] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[42] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3d reconstruction in function space," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4460–4470.

[43] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein, "Implicit neural representations with periodic activation functions," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7462–7473, 2020.

[44] E. Sucar, S. Liu, J. Ortiz, and A. J. Davison, "imap: Implicit mapping and positioning in real-time," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6229–6238.

[45] A. Rosinol, J. J. Leonard, and L. Carlone, "Nerf-slam: Real-time dense monocular slam with neural radiance fields," *arXiv preprint arXiv:2210.13641*, 2022.

[46] Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M. R. Oswald, and M. Pollefeys, "Nice-slam: Neural implicit scalable encoding for slam," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 786–12 796.

[47] D. Driess, J.-S. Ha, M. Toussaint, and R. Tedrake, "Learning models as functionals of signed-distance fields for manipulation planning," in *Conference on Robot Learning*. PMLR, 2022, pp. 245–255.

[48] M. Adamkiewicz, T. Chen, A. Caccavale, R. Gardner, P. Culbertson, J. Bohg, and M. Schwager, "Vision-only robot navigation in a neural radiance world," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4606–4613, 2022.

[49] M. Pantic, C. Cadena, R. Siegwart, and L. Ott, "Sampling-free obstacle gradients and reactive planning in neural radiance fields (nerf)," *arXiv preprint arXiv:2205.01389*, 2022.

[50] B. Lee, *Probabilistic Online Learning of Appearance and Structure for Robotics*. University of Pennsylvania, 2019.

[51] S. R. S. Varadhan, "On the behavior of the fundamental solution of the heat equation with variable coefficients," *Communications on pure and applied mathematics*, pp. 431–455, 1967.

[52] B. Lau, C. Sprunk, and W. Burgard, "Improved updating of euclidean distance maps and voronoi diagrams," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010.

[53] A. G. Belyaev and P. A. Fayolle, "On variational and PDE-based distance function approximations," *Computer Graphics Forum*, 2015.

[54] C. E. Rasmussen and C. K. Williams, *Gaussian Processes for Machine Learning*. Cambridge, Mass.: MIT Press, 2006.

[55] L. Liu, S. Fryc, L. Wu, T. L. Vu, G. Paul, and T. Vidal-Calleja, "Active and interactive mapping with dynamic gaussian process implicit surfaces for mobile manipulators," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3679–3686, 2021.

[56] S. Kim and J. Kim, *GPmap: A Unified Framework for Robotic Mapping Based on Sparse Gaussian Processes*. Springer International Publishing, 2015, pp. 319–332.

[57] J. Hartikainen and S. Särkkä, "Kalman filtering and smoothing solutions to temporal gaussian process regression models," in *2010 IEEE International Workshop on Machine Learning for Signal Processing*, 2010, pp. 379–384.

[58] P. Whittle, "On stationary processes in the plane," *Biometrika*, 1954.

[59] D. Eriksson, D. Kun, E. H. Lee, D. Bindel, and A. G. Wilson, "Scaling gaussian process regression with derivatives," *Neural Information Processing Systems (NIPS)*, 2018.

[60] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based slam," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2010.

[61] F. Bai, T. Vidal-Calleja, and G. Grisetti, "Sparse pose graph optimization in cycle space," *IEEE Transactions on Robotics*, vol. 37, no. 5, pp. 1381–1400, 2021.

[62] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz, "Towards 3d point cloud based object maps for household environments," *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 927–941, 2008.

[63] A. G. Wilson and H. Nickisch, "Kernel interpolation for scalable structured gaussian processes (kiss-gp)," *International Conference on Machine Learning*, pp. 1775–1784, 2015.

[64] K. Dong, D. Eriksson, H. Nickisch, D. Bindel, and A. G. Wilson, "Scalable log determinants for gaussian process kernel learning," *Neural Information Processing Systems (NIPS)*, pp. 6330–6340, 2017.

[65] E. H. W. Meijering, K. J. Zuiderveld, and M. A. Viergever, "Image reconstruction by convolution with symmetrical piecewise nth-order polynomial kernels," *IEEE Transactions on Image Processing*, 1999.

[66] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," in *Proc. of the Conf. on Computer Graphics and Interactive Techniques*, 1987.

[67] S. Anilkumar, "Lidar slam," https://github.com/soorajanilkumar/Lidar_SLAM, 2019.

[68] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp." in *Robotics: science and systems*, vol. 2, no. 4. Seattle, WA, 2009, p. 435.

[69] R. Kümmerle, B. Steder, C. Dornhege, M. Ruhnke, G. Grisetti, C. Stachniss, and A. Kleiner, "On measuring the accuracy of slam algorithms," *Autonomous Robots*, vol. 27, no. 4, pp. 387–407, 2009.

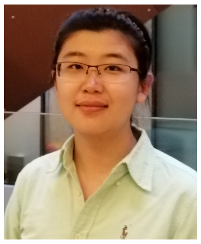
[70] A. Censi, "Scan matching in a probabilistic framework," in *Proc. of IEEE ICRA*. IEEE, 2006, pp. 2291–2296.

[71] G. Grisetti, C. Stachniss, S. Grzonka, and W. Burgard, "A tree parameterization for efficiently computing maximum likelihood maps using gradient descent." in *Robotics: Science and Systems*, vol. 3, 2007, p. 9.

[72] J. Sturm, W. Burgard, and D. Cremers, "Evaluating egomotion and structure-from-motion approaches using the TUM RGB-D benchmark," in *Workshop on Color-Depth Camera Fusion in Robotics at the IEEE/RJS Int. Conf. on Intelligent Robot Systems (IROS)*, 2012.

[73] M. Popović, T. Vidal-Calleja, G. Hitz, J. J. Chung, I. Sa, R. Siegwart, and J. Nieto, "An informative path planning framework for uav-based terrain monitoring," *Autonomous Robots*, vol. 44, p. 889–911, 2020.

[74] K. M. B. Lee, F. Kong, R. Cannizzaro, J. L. Palmer, D. Johnson, C. Yoo, and R. Fitch, "An upper confidence bound for simultaneous exploration and exploitation in heterogeneous multi-robot systems," in *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 8685–8691.



Lan Wu (Member, IEEE) received her B. Eng degree in electronic engineering in 2012, and her Ph.D. degree in robotics in 2023. Prior to joining the Robotics Institute (formerly Centre for Autonomous Systems, University of Technology Sydney) in 2019 for her Ph.D. degree, she worked in the industry as an electronic engineer, responsible for hardware and firmware designs for embedded systems. She is currently working as a postdoctoral research fellow at the Robotics Institute, University of Technology Sydney, Australia. Her research focus is probabilistic perception with depth sensors to perform efficient and effective mapping and accurate localisation for robotic systems.

Ki Myung Brian Lee (Member, IEEE) is currently a postdoctoral research fellow at the Robotics Institute, University of Technology Sydney (UTS), Ultimo, Australia. He completed his PhD at UTS in 2023, and his BEng at the University of Sydney in 2017. He is interested in Bayesian methods for robot perception with physics- or data-driven priors, and information-aware planning enabled by such perception models. More broadly, he is interested in mobile robot autonomy in previously unseen environments.

Cedric Le Gentil (Member, IEEE) received a DUT (associate degree) from Paris-Sud University (France) in 2012, and an M.Sc. from CentralSupelec (France) in 2015. He graduated with his PhD in robotics perception and state estimation in 2021 at the Robotics Institute of the University of Technology Sydney (Australia) where he is now a postdoctoral research fellow. Cedric's research interests revolve mostly around robotics perception and state estimation using various modalities: IMU, lidar, event cameras, ultrasound, etc. His past and present research includes multiple international collaborations and academic visits: the German aerospace centre (DLR) in Germany, the Autonomous Systems Lab at ETHZ in Switzerland, and the DREAM lab at GeorgiaTech Europe in France.

Teresa Vidal-Calleja (Senior Member, IEEE) received her BEng in mechanical engineering from the National Autonomous University of Mexico, in 2000, the MSc in mechatronics from CINVESTAV-IPN, Mexico, in 2002, and the PhD in automatic control, computer vision, and robotics from the Polytechnic University of Catalonia, Spain, in 2007. She was a Postdoctoral Research Fellow with LAAS-CNRS, France, and the Australian Centre for Field Robotics, the University of Sydney, Australia. In 2012, she joined the Robotics Institute at the University of Technology Sydney (UTS), Australia where she was a Chancellors Research Fellow and currently is Associated Professor and Research Director. She has been a Visiting Scholar with the Active Vision Laboratory, University of Oxford, U.K., with the Autonomous Systems Lab, ETH Zürich, Switzerland, and with the Institute of Robotics and Mechatronics of the German Aerospace Centre (DLR), Germany. Her research focus is on robotic perception combining estimation theory and machine learning.