Skeleton-based Conditionally Independent Gaussian Process Implicit Surfaces for Fusion in Sparse to Dense 3D Reconstruction

Lan Wu[1], Raphael Falque[1], Victor Perez-Puchalt[1,2], Liyang Liu[1], Nico Pietroni[1], and Teresa Vidal-Calleja[1]

***Abstract*— 3D object reconstructions obtained from 2D or 3D cameras are typically noisy. Probabilistic algorithms are suitable for information fusion and can deal with noise robustly. Consequently, these algorithms can be useful for accurate surface reconstruction. This paper presents an approach to estimate a probabilistic representation of the implicit surface of 3D objects. One of the contributions of the paper is the pipeline for generating an accurate reconstruction, given a set of sparse points that are close to the surface and a dense noisy point cloud. A novel submapping method following the topology of the object is proposed to generate conditional independent Gaussian Process Implicit Surfaces. This allows inference and fusion mechanisms to be performed in parallel to later propagating information through the submaps. Large datasets can efficiently be processed by the proposed pipeline producing not only a surface but also the uncertainty information of the reconstruction. We evaluate the performance of our algorithm using simulated and real datasets.**

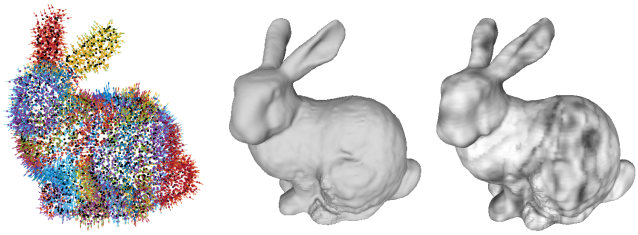
***Keywords*: Gaussian Process Implicit Surfaces, Skeleton Extraction, Bayesian Fusion, Conditionally Independent Maps.**

I. Introduction

Digitizing real-world objects for applications such as 3D printing requires accurate and high-resolution surface reconstruction. Although the use of 2D or 3D cameras makes this process efficient and cost-effective, it poses a challenge in the quality of the reconstructed surface. Object reconstruction from an image sequence is a key and well-studied problem in robotics and computer vision.

In the literature, there are a wide variety of methods for 3D reconstruction, a survey on this topic can be found in [1]. These methods vary in terms of the map representation, the data structure, the type of estimation, and other algorithmic differences, but only a few are probabilistic. Ref [2], for instance, is a well-known non-probabilistic open-source image to 3D reconstruction pipeline. It includes camera pose estimation, then multi-view stereo for dense map triangulation.

There are several contexts where probabilistic methods have demonstrated their effectiveness. For instance, due to its inherent uncertainty modelling, can highlight areas that might require. This representation can also guide active mapping approaches to revisit uncertain areas during the scanning processes. Another advantage of the probabilistic methods is the ability to fuse multiple sources of information. The popular Octomap approach [3], for instance, utilises occupancy probabilistic estimation to model the reconstructed object as part of a 3D grid (voxel grid). Also in a probabilistic manner, other approaches use Markov random field inference [4] to model the joint distribution over discrete occupancy and continuous appearance variables at each voxel given all input images. The problem in general with grid approaches is the need for discretisation, which hinders the ability to represent the surface properly.



(a) Input Data (b) Output Reconstruction (c) Output variance

Fig. 1. The proposed approach takes as input sparse 3D points reconstructed from visual SLAM methods and fuse it with noisy dense points obtained from depth or consecutive images. a) The noisy dense point cloud with normals is colored by submaps based on the proposed skeleton extraction algorithm. These dense points are used as training points for an efficient version of GPIS algorithm. In addition, an accurate sparse point cloud(in black) is fused with their specific conditionally independent GPIS (submap) and information is propagated across submaps. b) The final surface representation obtained by the algorithm given only as input a noisy point cloud with normals and accurate sparse points. c) Final reconstruction is colored by variance values from low (light grey) to high (dark grey) uncertainty.

Methods that leverage the space representation in a continuous manner can also be found in the literature. For instance, [5] proposes an approach for 3D shape reconstruction with images from a single camera based on Gaussian Process Latent Variable Models (GPLVM), which is a non-linear and probabilistic Gaussian Process based reduction technique.

Also exploiting Gaussian Processes (GP), [6], [7], [8] and [9] use Gaussian Process Implicit Surfaces (GPIS) allowing not only to deal with probabilistic information, but also to represent the uncertainty in the surface reconstruction. In this way, uncertainty information can be used for accurate grasping applications [6], to recover occupancy or meshes at arbitrary resolutions [8] or to improve the prediction of the surface. Any of these approaches integrate nicely to fusion algorithms.

The main issue with Gaussian Process-based algorithms, however, is the cubic computational complexity on the number of training points that prohibits applications for large-scale data. One solution to this computational constraint is the use of partitions a.k.a. "submaps". Prior work on mapping using submaps has been presented in the literature [10], [11]. The work in [12] exploits Conditional Independent (CI)

[1]All authors are with the Centre for Autonomous Systems, Faculty of Engineering and IT, University of Technology Sydney, Ultimo, NSW 2007, Australia. Corresponding author: Lan.Wu-1@student.uts.edu.au

[2]Victor Perez-Puchalt is also with the Swiss Federal Institute of Technology Lausanne, Ecole Polytechnique Federale de Lausanne, Lausanne 1015, Switzerland.

submaps and Bayesian fusion updates for 2.5D terrain mapping. We leverage this approach to reduce further the complexity of the already efficient GPIS method proposed in [9] for 3D object reconstruction using implicit surfaces. CI submapping in 2.5D representation requires partitions along one direction given the planarity of these maps. 3D surface reconstructions, however, require reasoning on the shape to partition the data in a meaningful way. In this paper, we propose the use of a skeletonization technique [13] for automatic partitioning of the 3D data following its hidden topology.

Other efficient and simple probabilistic representations such as sparse 3D points are the typical output of indirect methods for vision-based Simultaneous Localization and Mapping (SLAM) algorithms such as in [14]. The challenge is later to recover a dense probabilistic representation from the optimised sparse points. Our work here aims to tackle this specific problem; generating accurate probabilistic dense representations from a set of optimised sparse points and a set of noisy dense points with normals.

The approach proposed in this paper exploits a recently published efficient GPIS algorithm based on inducing points and gradients [15]. We use it as part of a skeleton-based CI submapping framework that takes advantage of Bayesian fusion to recover an accurate surface reconstruction (with uncertainty) given optimised sparse points and a dense noisy point cloud with normals. The proposed method also enforces continuity amongst the CI-submaps by using the Forward Propagation (FP) update proposed in [10], which we refer to as D-SKI-CI-Fusion. Fig. 1 shows the input and output of the approach and Fig. 2 presents its pipeline.

The remainder of the paper is structured as follows. Section II gives an overview of the proposed approach. Section III describes background on GPIS and the efficient approximation used. Then the approach, including the submapping based on skeletonisation and fusion algorithms, is detailed in Section IV. The results and evaluation are presented in Sections V. Finally, the conclusions are discussed in Section VI.

II. Approach Overview

Given a 3D reconstruction application from 2D or 3D cameras, the proposed approach takes as an input a set of sparse points Z (so-called keypoints) extracted from a feature-based visual SLAM algorithm and a point cloud with normals X obtained from multiple depth or consecutive images. It is assumed that Z is closer to the surface of the reconstructed object than X. This is a reasonable assumption as Z represents the points that have been triangulated during the SLAM estimation algorithm. The noise of X and Z is assumed to be normally distributed; thus Z has a smaller variance than X. Fig. 1.a) shows an example of the input and output of the approach in a simulated dataset.

Our probabilistic approach aims to estimate an accurate implicit surface with uncertainty information of the 3D object $p(\gamma|X,Z)$, where γ is the real implicit surface. We assume that γ is normally distributed. The devised approach consists of the following stages (refer to Fig. 2)):

1) A skeletonization algorithm is used to generate overlapping partitions based on Z following the object's topology. Then following these partitions, X is divided into n overlapping subsets.
2) An efficient GPIS model is then trained per subset using the corresponding points and normals X_{s_i}, and used to create the submaps s_i for $i = 1 \dots n$ at the query points $X^*_{s_i}$, producing a probabilistic implicit surface prior per submap, which is considered as the prior distribution $p(\gamma_{s_i}|X_{s_i})$. Note that $X^*_{s_i}$ are chosen by following the noisy point cloud normals to define points inside and outside the surface.
3) In parallel, the keypoints Z_{s_i} that belong to the s_i submap are fused with the corresponding prior implicit surface using the maximum a *posteriori* estimation $p(\gamma_{s_i}|X_{s_i}, Z_{s_i}) \propto p(\gamma_{s_i}|X_{s_i})p(Z_{s_i}|\gamma_{s_i})$.
4) The forward propagation algorithm proposed in [10] is applied to update the submaps $s_{1\dots n}$ in a sequential manner given the difference in the overlapping parts of the submaps.
5) Once all the keypoints have been fused and the information has been propagated to the last submap, the surface is recovered using both, the mean and covariance information of all the updated GPIS to recover the final $p(\gamma|X,Z)$.

III. D-SKI Based Gaussian Process Model

A. Gaussian Process

Gaussian Process (GP) [16] is a stochastic nonparametric machine learning technique that aims to represent a distribution over functions. The smoothing and generalisation information of the GP models are expressed in a kernel $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ and its hyperparameters $\boldsymbol{\theta}$ [16]. Once the kernel has been fully characterised using training points $X = \{\mathbf{x}_1, \dots, \mathbf{x}_{n_x}\}$, the $f(\mathbf{x}) \sim \mathcal{GP}(\boldsymbol{\mu}, P)$ model can be used to infer the probabilistic value at any arbitrary testing point X^* using,

$$\boldsymbol{\mu} = K(X^*, X)[K(X,X) + \sigma^2 I]^{-1}\gamma, \tag{1}$$

$$P = -K(X^*, X)[K(X,X) + \sigma^2 I]^{-1}K(X^*, X)^T + K(X^*, X^*), \tag{2}$$

where σ^2 is the variance of the additive independently distributed Gaussian noise of the input data, I the identity matrix and γ represents the input value. The matrix $K(X^*, X)$ is the $n_x \times n_{x^*}$ covariance matrix between the n_x training and n_{x^*} testing points.

Given a specific kernel, its hyperparameters $\boldsymbol{\theta}$ are learned using the maximum log-likelihood estimation to model the spatial correlation of input data. The log-likelihood

$$\log p(\gamma|X, \boldsymbol{\theta}) = -\frac{1}{2}(\gamma^T K_\gamma^{-1}\gamma + \log|K_\gamma| + n_x \log 2\pi), \tag{3}$$

where $|K_\gamma|$ represents the matrix determinant of $K_\gamma = K(X,X) + \sigma^2 I$ and $\frac{n_x}{2}\log 2\pi$ is a normalisation constant, with n_x as the number of training points.

B. Gaussian Process Implicit Surfaces with Derivatives

Formally, an implicit surface is defined as the surface described in Euclidean space which satisfies $f(\mathbf{x}) = 0$, where $\mathbf{x} \in \mathbb{R}^d$. In other words, the underlying surface is defined by the 3D points with null values.

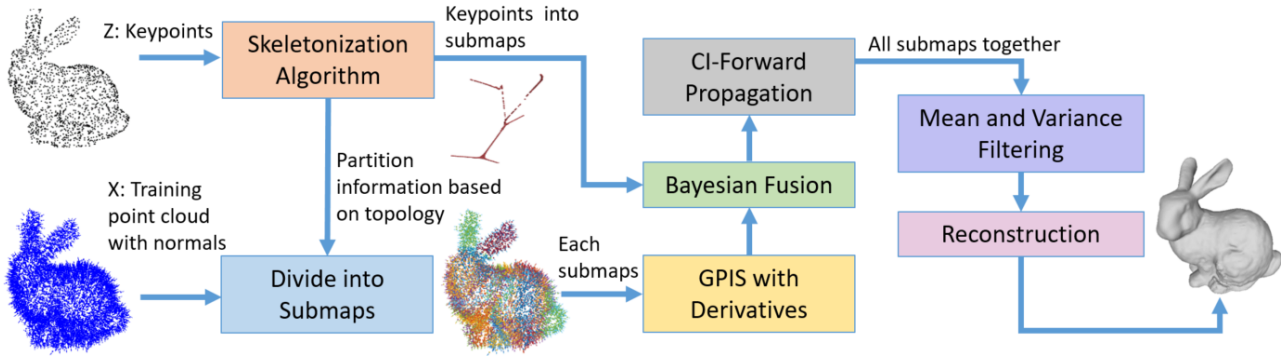


Fig. 2. Overall pipeline of the proposed approach.

Gaussian Process Implicit Surfaces [6], [17], [18], [7], [19], combine both notions, GP and implicit surfaces, in a continuous probabilistic representation of the 3D surface. In this work, as defined before, γ is the real implicit surface, and $p(\gamma|X)$ is the prior distribution of this surface defined by the noisy point cloud with normals $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_{n_x}\}$, which we model by a $\mathcal{GP}(\boldsymbol{\mu}, P)$.

GPIS with derivatives information [16], [19], [20] have been used to predict a more reasonably implicit surface by using the points and gradients during training, instead of adding arbitrary points with negative and positive values to represent the points inside and outside the surface respectively. In our case, the targeted implicit function is the null value on the surface with gradients, which are equivalent to the normals of the surface. Given that the derivative of a GP can also be regarded as a GP, the covariance function with derivatives can be written as following,

$$\begin{aligned} \operatorname{cov}\left(f(\mathbf{x}), \frac{\partial f(\mathbf{x}')}{\partial \mathbf{x}_i'}\right) &= -\frac{\partial k(\mathbf{x}, \mathbf{x}')}{\partial \mathbf{x}_i}, \\ \operatorname{cov}\left(\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}_i}, \frac{\partial f(\mathbf{x}')}{\partial \mathbf{x}_j'}\right) &= \frac{\partial^2 k(\mathbf{x}, \mathbf{x}')}{\partial \mathbf{x}_i \partial \mathbf{x}_j'}. \end{aligned} \tag{4}$$

Given n_x of d dimension with derivatives, computing the GP requires to solve an inversion of a positive definite matrix of dimension $n_x(d+1)$. Therefore, the computational complexity is $\mathcal{O}\left(n_x^3 d^3\right)$. In addition, the inference will cost $\mathcal{O}\left(n_x d\right)$ and $\mathcal{O}\left(n_x^2 d^2\right)$ per testing point.

C. Structured Kernel Interpolation with Derivatives (D-SKI)

To reduce GPIS model's computational complexity and improve its scalability, in this work, we proposed to use the scaling GP algorithm to achieve fast kernel learning [15], and prediction with derivatives information [9]. As pointed out before, given (3), there is a need to invert a matrix of the size of the training points. This becomes much worse when the derivatives are used as input. To avoid this cubic complexity, we have opted to use the recently proposed structured kernel interpolation framework with derivatives (D-SKI) [9]. D-SKI is based on the structured kernel interpolation method (SKI) [15] and the matrix-vector multiplications method (MVMs) [21] adding the derivatives as an extension. SKI uses an approximate kernel function with inducing points, enabling fast MVM by implementing interpolation to reproject the input dataset onto a grid generated by the inducing points,

$$K(X, X) \approx W K(U, U) W^T, \tag{5}$$

where U is a grid of inducing points m, and W is an $n_x \times m$ sparse matrix of interpolation weights.

If the input information involved with normals after an approximate kernel used, the GP will try to fit the function value and the gradients information separately, but this will cause the kernel no longer to be positive definite. D-SKI uses a differentiate approximate kernel function to solve this problem [9]. The target scaling kernel function with derivatives can be written

$$\begin{bmatrix} K & (\partial K)^T \\ \partial K & \partial^2 K \end{bmatrix} \approx \begin{bmatrix} W \\ \partial W \end{bmatrix} K(U, U) \begin{bmatrix} W \\ \partial W \end{bmatrix}^T = \begin{bmatrix} WK(U,U)W^T & WK(U,U)(\partial W)^T \\ (\partial W)K(U,U)W^T & (\partial W)K(U,U)(\partial W)^T \end{bmatrix}, \tag{6}$$

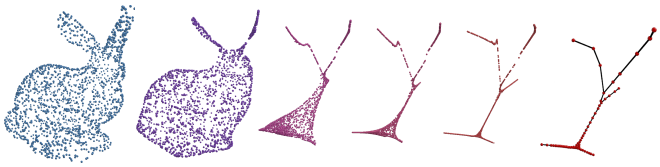
where W and ∂W are sparse matrices with assumption of quintic interpolation [22]. Note that the final computational complexity of D-SKI is $\mathcal{O}\left(n_x d 6^d + m \log m\right)$.

IV. Skeleton-based Conditionally Independent Submapping

Although D-SKI substantially reduces the complexity of GPIS, large 3D reconstructions are still intractable for most GP-based methods. In our previous work [12], CI submaps were proposed to address this problem. As pointed out in [12], each submap should meet the conditionally independence property to prevent discontinuities in the 3D reconstructions and to achieve an optimal map once the information has been propagated through the submaps. Note that the CI property is a desired condition that can be achieved by overlapping partitions arranged in a tree-like structure. Following this tree-like structure requirement, we proposed a skeletonization algorithm that partitions the data in a meaningful way, while respecting the CI property and maintaining similar submaps size.

A. Skeletonization Algorithm

While the submaps could be generated by dividing the input data into equal slices (*e.g.*, horizontal or vertical slices), this approach would not leverage the shape of the input data to create meaningful submaps, nor would it generate submaps with an equal number of points. This is particularly relevant when non-structured objects are reconstructed. As



(a) Keypoints (b) Point Cloud Contraction through Laplace-Beltrami operator (c) Skeleton

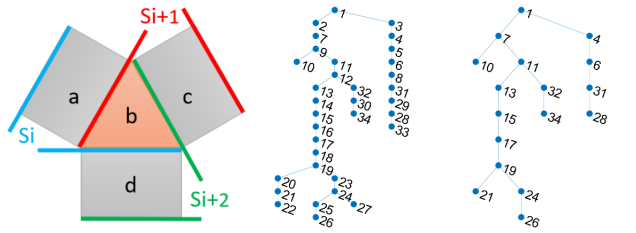
Fig. 3. Point cloud skeletonization: the keypoints a) are contracted by iterative application of the Laplace-Beltrami operator b). The contracted point cloud is then transformed into a tree using farthest distance sampling c).

an alternative, we consider the creation of submaps that follow the shape of the input data. Skeletonization algorithms naturally rise as a simple method for topologically traversing the data in a tree structure.

Amongst the skeletonization algorithm available in the literature [23], several methods have been specifically designed for point clouds. Amongst the popular methods, Tagliasacchi *et al.* [24] proposed a skeletonization method that can be used on incomplete point clouds where the points and the normals of these points are used to generate median axis keypoints. Later on, Cao *et al.* proposed to use a surface contraction based on Laplacian contraction [13]. As the latter method does not require normals, which in our case are initially not reliable, it is a good candidate for our application.

The method proposed by Cao *et al.* [13] relies on the Laplace-Beltrami operator, which is defined by building a local neighbourhood for each point of the point cloud. The Laplace-Beltrami operator is then used to iteratively contract the point cloud (following the idea from Au *et al.* [25]). Once the contraction's convergence is reached, a set of nodes is generated from the point cloud using *farthest point sampling*. The nodes are then connected by considering the connectivity through the local neighbourhood. This result in a graph which is then trimmed to generate a proper skeleton (see Fig. 3).

While building the nodes of the skeleton, the algorithm keeps track of the association between the nodes and the points from the point cloud. This information is then used to build the submaps. In contrast with Cao *et al.* [13], we propose to use k-nearest neighbours in the farthest distance sampling to generate submaps that have a similar number of points. This replacement is critical to ensure a constant computational time in the processing of a submap. Furthermore, in the trimming of the graph, we transform the skeleton graph into a tree as shown in Fig. 4.b) to be



(a) CI-submapping (b) Original and Final Skeleton

Fig. 4. The transformation of the skeleton into a submapping graph in b) is achieved by enforcing the conditional independent property shown in a).

able to propagate between submaps while respecting CI (as a consequence, CI submaping is naturally better suited for shape with *genus 0*).

B. CI-Submapping

The CI submaps are built by considering an overlapping section between two submaps to allow adjacent information propagation. The conditional independence property assumes that consecutive submaps s_i and s_{i+1} are independent given the overlapping section, therefore non-consecutive are strictly independent. An illustration of the CI submapping representation is shown in Fig. 4.a), in which the overall map S is divided into three submaps: s_i composed by $a+b$, s_{i+1} composed by $b+c$ and the last submap is conformed by $b+d$.

Splitting the data based on the proposed skeletonisation algorithm using Z and propagating the subdivisions to the noisy point cloud with normals, we obtain X_{s_i}. Then a GPIS is trained, and inference is performed on each $X*_{s_i}$ using D-SKI algorithm to obtain n CI GPIS submaps $\mathcal{GP}_{s_i}(\boldsymbol{\mu}_{s_i}, P_{s_i})$. Following our example on Fig. 4.a) the CI GPIS for s_i and the consecutive s_{i+1} are given by,

$$\boldsymbol{\mu}_{s_i} = \begin{bmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{bmatrix}, \qquad \boldsymbol{\mu}_{s_{i+1}} = \begin{bmatrix} \boldsymbol{\mu}_b \\ \boldsymbol{\mu}_c \end{bmatrix}, \tag{7}$$

$$P_{s_i} = \begin{bmatrix} P_a & P_{ab} \\ P_{ba} & P_b \end{bmatrix}, \qquad P_{s_{i+1}} = \begin{bmatrix} P_b & P_{bc} \\ P_{cb} & P_c \end{bmatrix}. \tag{8}$$

Note that the overlapping part is critical to deal with discontinuities related to applying independently per submap D-SKI-CI inference and fusion as discussed in the next section.

C. Forward Propagation

The FP algorithm allows the transfer of information from the s_{i-1}^{++} submap to the s_i, taking into account the difference on the overlapping part. Following [10] the FP step is as follows,

$$T = P_{cb}(P_b)^{-1} \tag{9}$$

$$P_{cb}^a = T P_b^a \tag{10}$$

$$P_c^a = P_c + T(P_{bc}^a - P_{bc}) \tag{11}$$

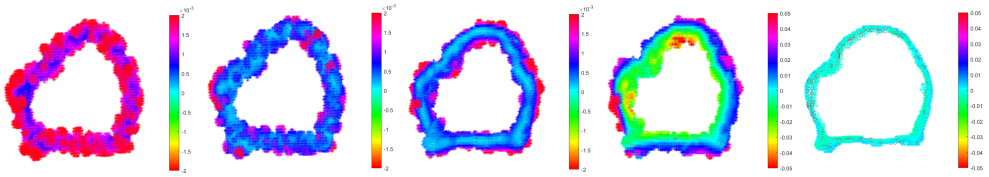
$$\boldsymbol{\mu}_c^a = \boldsymbol{\mu}_c + T(\boldsymbol{\mu}_b^a - \mu_b) \tag{12}$$

where the superscript corresponds to the information used to update the current submap. For instance, in the case of $\boldsymbol{\mu}_c^a$, the superscript represents the information coming from the non-common part of the previous submap, a, and the subscript c makes reference to the part of the submap that it's being updated.

In this way, Equations (11) and (12) are used to update the submap s_{i+1}, as follows:

$$\boldsymbol{\mu}_{s_{i+1}} = \begin{bmatrix} \boldsymbol{\mu}_b \\ \boldsymbol{\mu}_c^a \end{bmatrix}, \qquad P_{s_{i+1}} = \begin{bmatrix} P_b & (P_{cb}^a)^T \\ P_{cb}^a & P_c^a \end{bmatrix}. \tag{13}$$

The algorithm in general starts from the first submap s_1 and applies Bayesian fusion on the common and non-common parts, thus $s_1^+ = s_1^{++}$. Then FP is used to propagate the information to the next submap s_2 using the difference between the common part, updating only the non-common part of s_2. After this operation s_2 is updated s_2^+ and fusion



(a) (b) (c) (d) (e)

Fig. 5. Illustration of the implicit surface in a submap of the *Stanford Bunny* at given query points. a) σ^2 values after D-SKI. As moving away from the surface, in both directions outside-inside, the σ^2 increases. b) σ^2 values after D-SKI and fusion without forward propagation. Sparse to dense fusion reduces the uncertainty inside of submap, but the σ^2 values at the junction edge of this submap still need to be improved. c) After forward propagation and fusion, σ^2 values of certainty have significantly improved. d) μ values converge into the surface after FP and fusion used. It is very clear that points inside are with negative values and outside are with positive values. e) If the large values of σ^2 and μ have been taken from this submap, the result has some testing points around the surface.

is applied, thus becoming s_2^{++}. This process continues for all the submaps until the last submap s_n is updated by FP after the fusion process described in the following section.

D. Sparse to Dense Fusion

The Bayesian fusion step aims to incorporate the accurate information contained in the sparse keypoints Z into the prior D-SKI GPIS inference. By using the full covariance matrix, correlation is propagated across the whole submap, *i.e.*, a fusion step will not only update a single query point at the location of the sparse point, but also the neighbouring points. This allows us to improve the estimate of the posterior implicit surface of the i-th submap, even if the keypoints are very sparse, *i.e.*, $p(\gamma_{s_i}|X_{s_i}, Z_{s_i})$.

Given the GP prior and the keypoints, the maximum a *posteriori* estimation is used to update each submap with the keypoints lying in the submap. The update of the mean and covariance values of the CI GPIS in Equations (1) and (2) is performed as follows:

$$\boldsymbol{\mu}^+ = \boldsymbol{\mu} + PH^T(HPH^T + R)^{-1}(\gamma_Z - H\boldsymbol{\mu}) \tag{14}$$

$$P^+ = P - PH^T(HPH^T + R)^{-1}HP \tag{15}$$

with γ_Z representing the implicit surface value at the keypoints Z, H is a special observation matrix equal to one near the keypoint location and zero for the rest, and R a diagonal matrix representing the noise variance of the keypoints. Note that Bayesian fusion update is applied to each submap independently after each submap gets updated through the fusion step. For the sake of simplicity, we dropped the s_i subscript in the update equations.

Note that the D-SKI GPIS model trained per submap only requires $\mathcal{O}\left(n_x d6^d/n_s + m\log m\right)$, making the dense 3D reconstruction problem tractable. The fusion algorithm runs in parallel for the n_s submaps and it requires $\mathcal{O}((n_x^*/n_s)^3)$. FP runs in a sequential manner adding a negligible term because the inversion required in (9) is only on the number of points on the overlapping areas that are small by design. The total cost of our algorithm mainly depends on the number of inducing points, submaps and testing 3D points.

To summarize, the proposed approach gives the opportunity to process hundreds of thousands of dense noisy points, which is almost impossible to achieve effectively with standard GPs.

V. Evaluation

In this section, we present results first evaluating the D-SKI-CI-Fusion in simulated data and later we validate the approach with real data.

A. Implementation

Fig. 1.a) shows the dataset used for the evaluation of the approach. The 30mm *Stanford Bunny*[1] dataset is down-sampled by 20% (as per [9]) and $\sigma_X = 0.05cm$ noise is added to emulate the noisy dense point cloud and only 5% of the original Bunny dataset with no noise is used to emulate the keypoints. The full approach was implemented in Matlab and ran on an Intel i7-8650U CPU@1.9GHz.

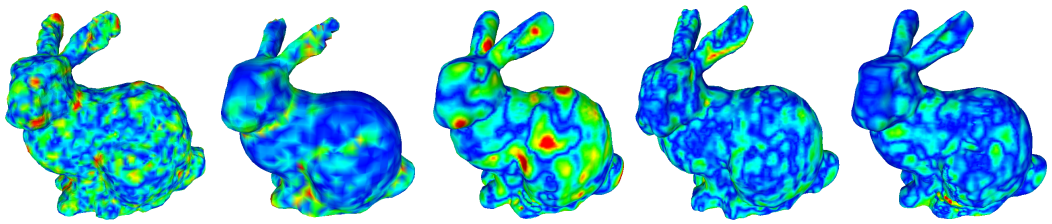
As an illustration of the whole pipeline in a cross-section, Fig. 5 shows the D-SKI-CI-Fusion obtained at the query points $X^*_{s_5}$ in the submap obtained by the skeletonization algorithm described in Section IV, using the noisy data of the *Stanford Bunny* provided in Fig. 1.a). The underlying surface, represented by the keypoints Z_{s_5} at this cross-section are shown as small red circles. Note that the accuracy of the GPIS model depends on the noise of the initial dense point cloud and could cause large errors even after GPIS inference. The ability to combine mean and variance information allows filtering the large errors from the mean prediction, generating a better approximation of the real surface. Following our cross-section example, Fig. 5.e) shows the filtered prediction that lies near the keypoints.

We only use query points within a distance radius of d (based on the point cloud variance) of the training points. This guarantees that the training points are near the surface, reducing the complexity of querying in a regular grid. Every submap is normalised based on the size of the enclosing area to aid the convergence of the training algorithm. After inference, the re-scaling factor used to normalise the input is applied back. Note that we use the Thin Plate kernel [7] with D-SKI approximation.

B. Error Evaluation

A qualitative evaluation of the surface reconstruction with Poisson algorithm is presented Fig. 6. The five reconstructions display the distance from the vertex of the Poisson reconstruction to the closest point in the original triangular mesh. The reconstruction from the initial noisy data X is shown in Fig. 6.a). From the figure, we can observe an

[1] http://graphics.stanford.edu/data/3Dscanrep/.



(a) (b) (c) (d) (e)

Fig. 6. Comparison of the input and output surface reconstructions, coloured by distance error. a) Reconstruction from initial sensor dataset, b) only use keypoints to reconstruct surfaces, c) D-SKI-CI without forward propagation and fusion, d) D-SKI-CI-Fusion with linear partition instead of skeletonization partition, e) D-SKI-CI-Fusion after forward propagation and fusion. The surface reconstructions were obtained by Poisson algorithm using the same parameters.

inaccurate mesh due to the induced noise, even though Poisson inherently filtered some of it. The reconstruction using only keypoints as input to Poisson is shown in Fig. 6.b). The smooth surface is lacking details, resulting in a less accurate reconstruction compared to our proposed method in Fig. 6.e). The D-SKI with skeletonization CI submapping right before FP and fusion is shown in Fig. 6.c). The discontinuity between submaps is visible, resulting in artefacts caused by Poisson on the bunny's back. Fig. 6.d) shows D-SKI-CI-Fusion using a linear partition along the horizontal axis instead of skeletonization. This linear partitioning causes larger errors in the ears, possibly due to the hyperparameters fitting, *i.e.,* the ear correlations are different from the body. The final reconstruction shows a reduction of the error across the whole body as the partitions based on topology help hyperparameters fitting (ears are in different submaps than the body). This is further reduced by fusion and forward propagation steps updating the initial D-SKI-CI.

Given that this dataset is not too large, the computation of the standard GPIS algorithm [7] with no derivatives and no submaps is actually feasible. GPIS takes 33 minutes, while our algorithm achieves a 35% reduction in the final computation time with the same accuracy.

To study the influence of the point cloud noise on the quality of our algorithm's final result, we perform a test varying the amount of noise added to the ground-truth. To show the validity of our algorithm, we compute the minimum distance from each point of the prediction to the real surface. For the root mean square error (RMSE) evaluation, the final prediction is downsampled to the same resolution as the initial dataset, which allows for a fair comparison.

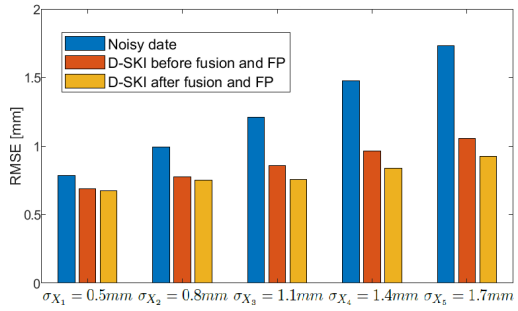
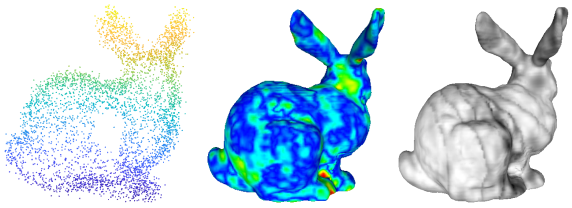


Fig. 7. Comparison of RMSE for different approaches: initial data, D-SKI before FP and fusion and D-SKI after FP and fusion implemented.

Fig. 7 shows the RMSE over all the input and predicted points (before and after fusion) with the real surface. The distance is computed from point to the closest point on the initial mesh of the ground truth. From left to the right colored bars, the correspondences are the initial noisy dataset, the D-SKI prediction using the noisy dataset without fusion and our final approach D-SKI after FP and fusion. It is clear that the GPIS algorithm filters the most amount of noise, keeping it constant even with an increased noise variance. Note the error reduction ranges from 8% to 32% given an increment of noisy of $\sigma_X = 1.2$mm.

C. Missing Data

SLAM algorithms might produce point clouds with missing data due to areas with lack of texture or surface's material. Fig 8.a) show simulated missing data in the dense point cloud and the reconstructed surface applying our approach in Fig 8.b) coloured by error. Fig 8.c) shows the variance, something to point our here is the low variance in the missing data. This is due to the fusion step with the keypoints, that makes the variance low given this accurate information. Other regions that require attention are highlighted here by the variance such as the back paw, motivating the use of active mapping algorithms.



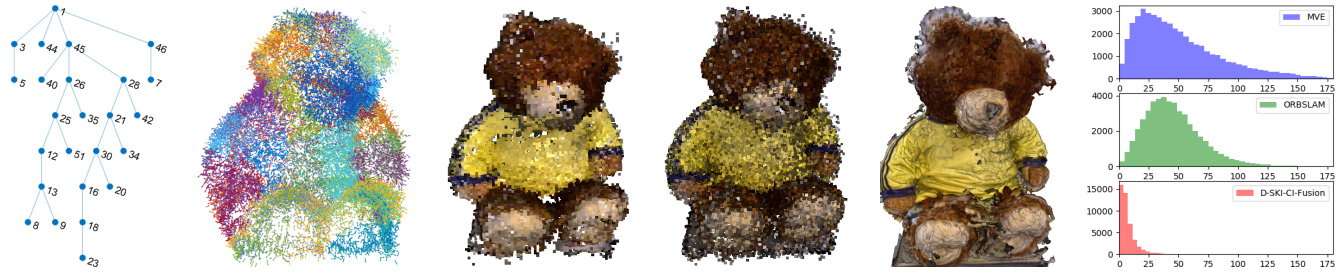
(a) Noisy dense points (b) Distance error (c) Variance

Fig. 8. Surface reconstruction from partial leg missing. a) Point cloud, b) coloured by distance to the ground-truth and c) coloured by variance.

D. Experimental Validation

To test the D-SKI-CI-Fusion algorithm in real and more challenging scenarios, data from the TUM dataset [26], the large scale Jenolan caves [27], and an additional in-house dataset representing the carcass of a lamb were used.

From the TUM dataset, we chose the challenging (low texture) sequence of "Freiburg3 Teddy", which was acquired with an RGBD camera and reconstructed using both



(a) (b) (c) (d) (e) (f)

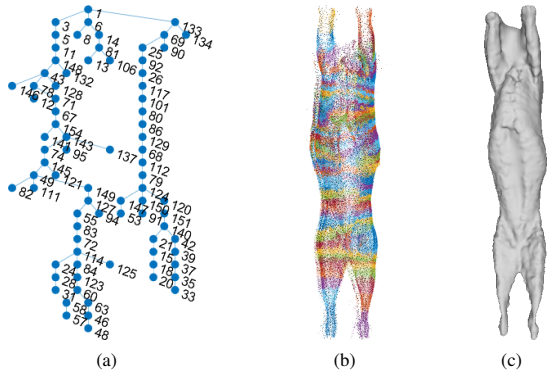
Fig. 9. Surface reconstructions for *Teddy Bear* dataset. a) Submapping graph, b) initial noisy point cloud by coloured submaps, c) D-SKI-CI-Fusion, d) ORB-SLAM, e) MVE mesh, f) histogram of the angles between smoothed and original normals (in degree).

ORB-SLAM [14] and Multi-View Environment (MVE) algorithm [2] (RGB only). ORB-SLAM camera poses and keypoints (ORB features) are optimised in an online setup and a post-processing step computes the dense point reconstructions using the original depth and the optimised camera poses. MVE, on the other hand, after the optimisation computes an implicit surface as a multi-scale hierarchical signed distance field volumetric representation (stored in an octree). It uses marching cubes to extract the zero-level set to provide the mesh. We aim to benchmark the proposed approach against these two reconstruction methods in a qualitative and quantitative manner. We use ORB-SLAM keypoints and dense points to performed D-SKI-CI-Fusion.

Given the keypoints Z and the dense noisy point cloud X from ORB-SLAM, we applied the proposed D-SKI-CI-Fusion approach to recover a denoised point reconstruction. The results are shown in Fig. 9.c). As a qualitative comparison, the results of ORB-SLAM and MVE are shown in Fig. 9.d) and 9.e) respectively. Note that the skeleton CI mapping, the noisy point cloud submapped are shown in the subfigures a) and b) respectively. As a quantitative evaluation, we iteratively average the ten closest neighbours to smooth the normals and get the histogram of the angles between the orginal and smoothed normal for each reconstruction. The histograms in Fig. 9.f) show that our method produces normals with less noise than ORB-SLAM and MVE.

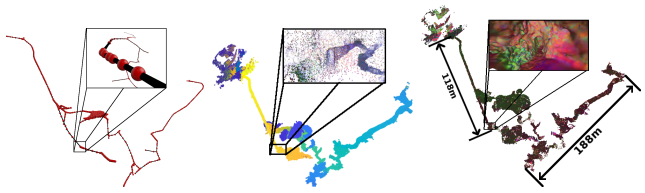
Secondly, we have acquired lamb carcass images in a chiller, in which the body was hanging on a rail. A multiple 3D cameras system have scanned the whole carcass at small increments. A Visual-SLAM algorithm [28] was used to reconstruct the 3D carcass with camera poses and sparse points. This algorithm does not produce a dense reconstruction and MVE failed to recover a meaningful surface. The sparse points from the visual-SLAM algorithm were used as the keypoints Z. The dense data was obtained by recovering the noisy 3D points directly from the depth images at a set poses output of the SLAM algorithm.

The reason why MVE failed is due to labels attached to the carcass which were moving with the ventilation of the room. This unstable environment makes a challenging example in which the data collected from the sensors will not be accurate, and in some case, a double surface representation can be found. The results of the skeletonisation algorithm, noisy point cloud and the reconstruction after D-SKI-CI-Fusion with Poisson are shown in Fig. 10.



(a) (b) (c)

Fig. 10. D-SKI-CI-Fusion method applied to *Lamb carcass*. a) Submapping graph, b) initial noisy point cloud colored by submaps, c) D-SKI-CI-Fusion surface reconstruction with Poisson algorithm.



(a) (b) (c)

Fig. 11. D-SKI-CI-Fusion method applied to the *Jenolan Caves* dataset. Zoomed in regions show a detail of the caves. a) Skeleton, b) initial noisy point cloud colored by submaps, c) D-SKI-CI-Fusion final point cloud.

Finally, in order to show that our algorithm can cater for larger amounts of data we have included the Jenolan caves dataset [27], which contains more than one million points. Fig. 11.a) shows the skeleton of this data. Note that although this dataset contains loops, our skeletonization approach can deal with them to produce a CI partition arranged on a tree structure. Fig. 11.b) shows the original point cloud coloured by submaps and the final reconstruction after applying Poisson algorithm is shown in Fig. 11.c).

Table I shows the number of points on the sensor data, the keypoints, median points per submap and final points were utilised as part of the algorithm for the Freiburg3 Teddy, lamb carcass, Jenolan caves and simulated datasets. The median submap time is also shown here. This time considers all steps except for skeletonization. Note that some submaps size is variable. It is clear from the table that carcass and Jenolan

TABLE I

DATASETS POINTS AMOUNTS AND RUNNING TIME

	X	Median X_{s_i}	X^*	Median $X^*_{s_i}$	Z	Median Z_{s_i}	$N_{submaps}$	Average Time/s_i
Stanford Bunny	6,964	602	77,123	4,587	1,741	138	17	80.952 s
Freiburg Teddy	17,703	968	65,261	2,665	9,530	568	25	105.9 s
Lamb carcass	169,317	2,809	727,247	8,582	7,028	129	81	222.372 s
Jenolan Caves	1,009,630	3,750	4,554,780	11,250	252,407	963	357	247.284 s

caves datasets are quite challenging in terms of the amount of data managed and fused.

VI. CONCLUSIONS

This paper presents a probabilistic approach for recovering dense 3D reconstructions from a set of optimised 3D points and a noisy point cloud with normals. The algorithm, first subdivides the dense point cloud into a Conditionally Independent submaps following the shape of the object. Note that for objects with structured shapes such as rooms or tables, simple linear partitions will produce by the algorithm. After partitioning the data, an efficient GPIS algorithm is used to predict a probabilistic implicit surface that after filtering is fused with optimised points commonly provided by a feature-based SLAM algorithm. The forward propagation algorithm is finally used to correlate the submaps improving the inference.

Our approach is capable to deal with large-scale data and reconstruct an accurate surface that contains uncertainty information, giving the ability to highlight issues on the final reconstruction.

Note that the devised approach could also work without the optimised points, to simply filtered the noisy point cloud or to fuse sequential noisy data. Future work considers extending this algorithm to recover occupancy information, which provides with the ability to be used beyond object reconstruction.

ACKNOWLEDGEMENT

This research is supported by an Australian Government Research Training Program Scholarship. The authors would like to thank Alen Alempijevic and Phillip Quin for providing the lamb carcass dataset.

REFERENCES

[1] M. Zollhofer, P. Stotko, A. Gorlitz, C. Theobalt, M. Niebner, R. Klein, and A. Kolb, "State of the art on 3d reconstruction with rgb-d cameras," *Computer Graphics Forum*, vol. 37, pp. 625–652, 05 2018.

[2] 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.

[3] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: an efficient probabilistic 3d mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, 2013.

[4] A. O. Ulusoy, A. Geiger, and M. J. Black, "Towards probabilistic volumetric reconstruction using ray potentials," in *International Conference on 3D Vision*, 2015, pp. 10–18.

[5] V. A. Prisacariu, A. V. Segal, and I. Reid, "Simultaneous monocular 2d segmentation, 3d pose recovery and 3d reconstruction," in *Asian conference on Computer Vision (ACCV)*. Springer, 2012, pp. 593–606.

[6] S. Dragiev, M. Toussaint, and M. Gienger, "Gaussian process implicit surfaces for shape estimation and grasping," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 2845–2850.

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

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

[9] 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.

[10] L. Sun, *Large-Scale Continuous 2.5D Robotic Mapping*. Center of Autonomous Systems, University of Technology Sydney, 2018.

[11] S. Kim and J. Kim, "Occupancy mapping and surface reconstruction using local gaussian processes with kinect sensors," *IEEE Transactions on Cybernetics*, vol. 43, no. 5, pp. 1335–1346, 2013.

[12] L. Sun, T. Vidal-Calleja, and J. Valls Miro, "Bayesian fusion using conditionally independent submaps for high resolution 2.5d mapping," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 3394–3400.

[13] J. Cao, A. Tagliasacchi, M. Olson, H. Zhang, and Z. Su, "Point Cloud Skeletons via Laplacian Based Contraction," *Shape Modeling International Conference*, pp. 187–197, 2010.

[14] 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.

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

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

[17] G. Turk and J. F. OBrien, "Variational implicit surfaces," *Georgia Institute of Technology*, 1999.

[18] S. Caccamo, Y. Bekiroglu, C. H. Ek, and D. Kragic, "Active exploration using gaussian random fields and gaussian process implicit surfaces," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 582–589.

[19] W. Martens, Y. Poffet, P. R. Soria, R. Fitch, and S. Sukkarieh, "Geometric priors for gaussian process implicit surfaces," *IEEE Robotics and Automation Letters (RAL)*, vol. 2, no. 2, pp. 373–380, 2017.

[20] S. Kim and J. Kim, "Hierarchical gaussian processes for robust and accurate map building," *Australasian Conference on Robotics and Automation (ACRA)*, pp. 117–124, 2015.

[21] 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.

[22] 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*, pp. 192–201, 1999.

[23] A. Tagliasacchi, T. Delame, M. Spagnuolo, and N. Amenta, "3D Skeletons : A State-of-the-Art Report," *Computer Graphics Forum*, vol. 35, no. 2, pp. 573–597, 2016.

[24] A. Tagliasacchi, H. Zhang, and D. Cohen-Or, "Curve skeleton extraction from incomplete point cloud," *ACM Transactions on Graphics*, vol. 28, no. 3, p. 1, 2009.

[25] O. K.-C. Au, C.-L. Tai, H.-K. Chu, D. Cohen-Or, and T.-Y. Lee, "Skeleton extraction by mesh contraction," *ACM Siggraph*, p. 1, 2008.

[26] 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 International Conference on Intelligent Robot Systems (IROS)*, 2012.

[27] R. Zlot and M. Bosse, "Three-dimensional mobile mapping of caves," *Journal of Cave & Karst Studies*, vol. 76, no. 3, 2014.

[28] S. Rahman, P. Quin, T. Walsh, T. Vidal-Calleja, M. McPhee, E. Toohey, and A. Alempijevic, "Preliminary estimation of fat depth in the lamb short loin using a hyperspectral camera," *Animal Production Science*, vol. 58, pp. 1488–1496, 2018.