

Active Exploration Using Gaussian Random Fields and Gaussian Process Implicit Surfaces

Sergio Caccamo, Yasemin Bekiroglu, Carl Henrik Ek, Danica Kragic

Abstract—In this work we study the problem of exploring surfaces and building compact 3D representations of the environment surrounding a robot through active perception. We propose an online probabilistic framework that merges visual and tactile measurements using Gaussian Random Field and Gaussian Process Implicit Surfaces. The system investigates incomplete point clouds in order to find a small set of regions of interest which are then physically explored with a robotic arm equipped with tactile sensors. We show experimental results obtained using a PrimeSense camera, a Kinova Jaco2 robotic arm and OptoForce sensors on different scenarios. We then demonstrate how to use the online framework for object detection and terrain classification.

Index Terms—Active perception, Surface reconstruction, Gaussian process, Implicit surface, Random field, Tactile exploration.

I. INTRODUCTION

Acquiring a high quality 3D model of the environment is critical for many autonomous robotics problems such as grasping, segmentation, traversability or mapping. Mere vision perception does not often exhaustively describe the shape of the environment since volumetric data generated from modern vision sensors are prone to errors due to limited field of view, photometric effects, occlusions and noise.

Passive observation of a scene leads to incomplete shapes of objects and terrains facing the camera. Heuristic or symmetry assumptions [1] can be used to deal with lack of data in the observations, leading to errors in the representation.

Surface exploration through vision and haptic interactions is the task of purposefully touch and inspect a portion of environment so to reveal occluded information. It can be considered as a case of either interactive or active perception depending whether the physical interaction strategically modify the environment under analysis or not respectively.

Haptic exploration helps improving observations adding a new layer of information into the world model. Meier et al. [2] showed that tactile information alone can be used to adequately describe objects properties. A robotic manipulator equipped with tactile sensors can be used to encode properties of surfaces and objects and enhance visual perception of shapes [3]. Studies [4] show that combining tactile and visual representations of object brings more reliable and robust shape estimation than either the visual or tactile alone.

Caccamo, Ek and Kragic are with the Computer Vision and Active Perception Lab., Centre for Autonomous Systems, School of Computer Science and Communication, Royal Institute of Technology (KTH), SE-100 44 Stockholm, SE. e-mail: {caccamo|chek|dani}@kth.se

Bekiroglu is with the School of Mechanical Engineering, University of Birmingham, UK e-mail: {Y.Bekiroglu}@bham.ac.uk

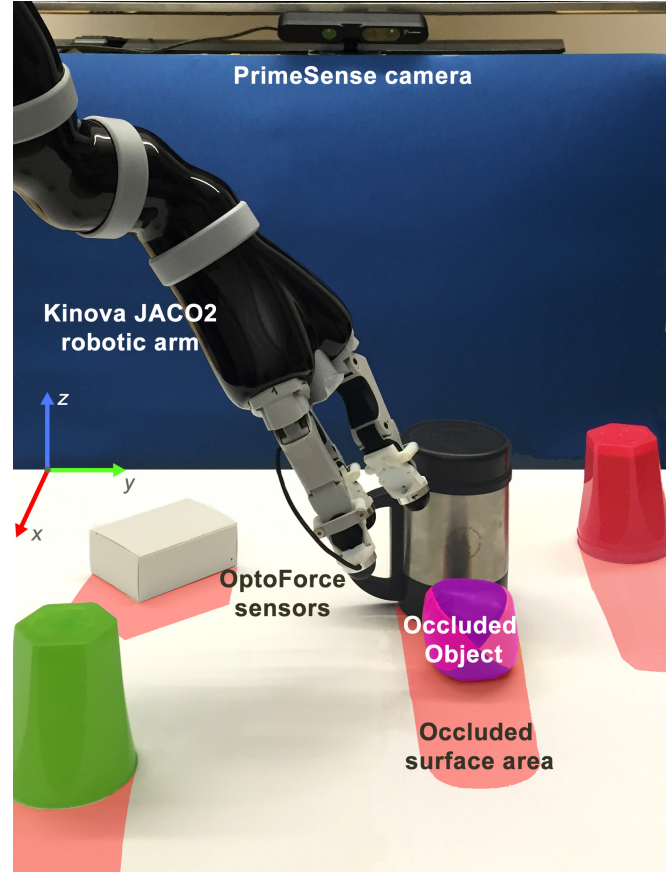


Fig. 1: Experimental setup. A kinova Jaco2 arm equipped with OptoForce sensor explores occluded areas of an environment.

Even humans build their world representation using different sensory information and actively examine the environment to enhance their perception [5].

In this paper we use the term VRS (visible and reachable surface) to refer to all the surfaces contained in the portion of space consisting in the intersection of the field of view of the camera and the reachable space of the robot. The VRS can contain several occluded regions and objects (i.e. occluded surfaces, see Fig. 1).

Examples of VRS are:

- The surface of a table and objects placed in front of a robot.
- The portion of map in front of an arm equipped Unmanned Ground Vehicle (UGV) as shown in Fig. 2.

Inspired by the work in [3] and field applications described in [6] we build 3D models of a VRS by merging vision and

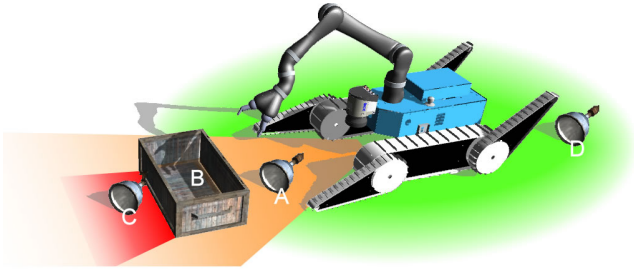


Fig. 2: Visible and reachable surface of an UGV. Objects A and B are inside the VRS, B is partially occluded. C and D are outside the VRS.

haptic data into a probabilistic framework. We study how to properly model shape of environments that contain different occluded objects and incomplete areas taking advantage of uncertainties in the sensory system. We also show how to reduce the exploration time and the amount of physical interactions needed.

In summary, the contributions of this paper can be listed as follow:

- 1) We propose a new probabilistic framework for surface exploration that merges haptic and visual sensory information for building a local 3D map of the environment.
- 2) We show how to exploit Gaussian Processes and Delaunay triangulation for reducing the amount of interactions and the computational power/time needed.
- 3) We demonstrate the feasibility of the approach through hardware experiments letting a robotic arm explore different scenarios and show how few interactions can add useful information to a partially visible surface.
- 4) We show how to generically exploit the autonomous framework for problems such as object identification and terrain classification.

II. RELATED WORK

Gaussian processes (GP) [7] have been used for terrain mapping and modeling [8], [9] in a wide range of applications including geophysics, aeronautics and robotics. In those studies any given point in a 2D Euclidean space is associated with a single elevation value, generating therefore a 2.5D surface called digital elevation model (DEM) also known as heightmap [10]. Another example of Gaussian processes applied to digital terrain modeling can be seen in [11]. Such models may fit terrain shapes but are not suitable for more complex surfaces or in applications such as object manipulation or segmentation.

Implicit surfaces [12] have been widely used to represent object shapes since their first appearances in [13]. Generic reconstruction of implicit surfaces from data points have been presented in [14]. Machine learning techniques have been progressively developed to represent complex surfaces as in [15] and [16]. More recently, Gaussian Process Implicit Surfaces (GPIS) [17] have become very popular allowing to extend implicit surface to uncertainty, a property needed when the model is the result of sensory data fusion [18].

Environmental observations can condition a GP so that its posterior mean define the implicit surface defining the terrain (including objects). Authors in [19] applied GPIS for building 3D representation of the environment by fusing laser and mm-wave radar data. Results in [20] show how GPIS as object representation can even improve complex tasks as grasping.

One disadvantage of these approaches is that during inference, Gaussian Process Regression (GPR) is computationally demanding. The major cost takes place from the inversion of large covariance matrices that, in the simplest implementation, have complexity $O(n^3)$. Mathematical tools as Cholesky decomposition or sparse kernels [21] can considerably reduce the computational effort.

GPIS requires a dense cubic matrix of points as test set in order to qualitatively describe the implicit surface. When a VRS includes several objects, the matrix containing the training sample points (i.e. the point cloud) becomes large and the computational time increases drastically making the implementation of an online active perception algorithm for surface exploration infeasible. To overcome such problem a down-sampled subset is often heuristically selected and used [22].

An application of sparse kernels for mapping of large area is presented in [23] where authors propose a unified framework for building continuous occupancy grid maps. As already mentioned, merging haptic and visual sensory data into the same probabilistic model using GPs can lead to better shape representation [3] and planning [24]. An example of tactile sensing for object tracking with visual occlusion using particle filters is presented in [25]. Another efficient tactile perception algorithm for object manipulation and tracking is shown in [26].

Work in [27] and [28] showed that the morphology of an environment is very important for an unmanned ground vehicle (UGV) that autonomously tries to traverse a particularly harsh environment. They showed also that occlusion and reflections, e.g. caused by a pot of water or broken glass, can lead to failure. This could be solved by asking a robotic arm to strategically explore the environment around the UGV. Nevertheless, the battery capacity, the computational power on board of the robot and limited time constraints, common during urban search and rescue missions, force the exploration task to reduce as much as possible the required elaboration time and the tactile interactions. On a different scenario, an interactive humanoid robot, which explore a table with objects for segmentation, can try to identify hidden elements by touching occluded regions and move its head only in case of positive tactile feedback.

To address those problems we propose a probabilistic method that identifies and analyses occluded regions in the working space area of a robot, where a VRS point cloud can be much larger than a single object (e.g. cup or bottle). We train a Gaussian Random Field (GRF) and a Gaussian Process Implicit Surface on the initial point cloud representing the VRS. We then infer the joint distribution of Gaussian Random Field model on regions of interest obtained from

a 2.5D fast Delaunay triangulation. Delaunay triangulation on a discrete Euclidean d-dimensional point set corresponds to the dual graph of the Voronoi diagram [29] for the same set. We use it to quickly identify and investigate large sparse areas in the visual point cloud that could potentially carry high uncertainty in the internal probabilistic model. We use a robot manipulator with tactile sensors for autonomously touching the isolated regions of the surface. We define a new training set for the GPIS using on-surface and off-surface tactile points obtained during each interaction by tactile sensors placed on the fingertips of the robotic hand. Finally we generate the new 3D shape inferring the GPIS on a subset of the VRS selected using the predicted mean of the GRF.

A surface exploration step in this paper denotes a single iteration of the algorithm that includes several physical interactions with the environment. During each exploration step the GPIS model is updated many times enlarging its training set with tactile information. We assume the environment to remain static during the whole analysis.

III. SURFACE MODELING

In this section we briefly discuss Gaussian Processes Regression (GPR) [7]. We describe how to exploit two dimensional GPR (Gaussian Random Field) and three dimensional GPR (Gaussian Process Implicit Surfaces [17]) for modeling terrain and object shapes.

A. Gaussian Random Fields

We denote $P_{VRS} = \{\mathbf{x}_1, \mathbf{x}_2 \dots \mathbf{x}_N\}$ with $\mathbf{x}_i \in \mathbb{R}^3$ the set of measurements of 3D points lying on the visible reachable surface and $D_{RF} = \{\mathbf{x}_i, y_i\}_{i=1}^N$ a bi-dimensional training set where $\mathbf{x}_i \in \mathbf{X} \subset \mathbb{R}^2$ are the xy-coordinates of the points in P_{VRS} and y_i the z-coordinates (heights)¹. We also define a set $\mathbf{X}_* \equiv \mathbf{X}_{rf*} \subset \mathbb{R}^2$ of M test points. With a function $f: \mathbb{R}^2 \rightarrow \mathbb{R}$ we map a 2.5D surface where each vector of xy-coordinates generates a single height. Such a function can efficiently be modeled by a GPR which places a multivariate Gaussian distribution over the space of $f(\mathbf{x})$. The GP can be fully described by a mean function $m(\mathbf{x})$ and a covariance function $k(\mathbf{x}_i, \mathbf{x}_j)$. Assuming noisy observation $y = f(\mathbf{x}) + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$ and $m(\mathbf{x}) = 0$ the joint Gaussian distribution on the test set \mathbf{X}_* becomes

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} \mathbf{K} + \sigma_n^2 \mathbf{I} & \mathbf{k}_* \\ \mathbf{k}_*^T & k_{**} \end{bmatrix}\right) \quad (1)$$

where \mathbf{K} is the covariance matrix between the training points $[\mathbf{K}]_{i,j=1\dots N} = k(\mathbf{x}_i, \mathbf{x}_j)$, \mathbf{k}_* the covariance matrix between training and test points $[\mathbf{k}_*]_{i=1\dots N, j=1\dots M} = k(\mathbf{x}_i, \mathbf{x}_{*j})$ and k_{**} the covariance matrix between the test points $[k_{**}]_{i,j=1\dots M} = k(\mathbf{x}_{*i}, \mathbf{x}_{*j})$.

The predictive function is obtained by conditioning on the training points

$$p(f_* | \mathbf{X}_*, \mathbf{X}, \mathbf{y}) = \mathcal{N}(\bar{f}_*, \mathbb{V}[f_*]) \quad (2)$$

$$\bar{f}_* = \mathbf{k}_*^T (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{y} \quad (3)$$

$$\mathbb{V}[f_*] = \mathbf{k}_{**} - \mathbf{k}_*^T (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{k}_* \quad (4)$$

For this study we choose to use the popular squared exponential kernel

$$k(\mathbf{x}_i, \mathbf{x}_j) = \sigma_e^2 \exp\left(-\frac{(\mathbf{x}_i - \mathbf{x}_j)^T (\mathbf{x}_i - \mathbf{x}_j)}{\sigma_w^2}\right) \quad (5)$$

Gaussian random field (GRF) is a common way to refer to Gaussian Process Regressors that generalize over bi-dimensional Euclidean vectors. Associating every coordinate to a single height is a big limitation when it comes to represent complex surfaces, e.g. mugs, inclined boxes. On the other hand, inferring a random field will directly produce 3D points by combining input-output into vectors of coordinates. This explicit behavior of the joint distribution permits to quickly obtain a DEM querying large portion of the VRS using only few bi dimensional testing points. The variance of the random field allows to directly highlight regions of low density data, e.g. occluded portion of the VRS, or high complexity portion of surface, e.g. different heights for the same coordinate.

B. Gaussian Processes Implicit Surfaces

Gaussian Process Implicit Surface (GPIS) models a function $f: \mathbb{R}^3 \rightarrow \mathbb{R}$ with supporting points defining an implicit surface. Whereas equations 1, 2, 3, 4 maintain the same form, $D_{IS} = \{\mathbf{x}_i, y_i\}_{i=1}^N$ becomes the new training set where $\mathbf{x}_i \in \mathbf{X} \subseteq P_{VRS}$ and $y_i \in \mathbb{R}$ defined as in [17]

$$y_i \begin{cases} = -1, & \text{if } \mathbf{x}_i \text{ is below surface} \\ = 0, & \text{if } \mathbf{x}_i \text{ is on the surface} \\ = 1, & \text{if } \mathbf{x}_i \text{ is above the surface} \end{cases} \quad (6)$$

We also redefine the set $\mathbf{X}_* \equiv \mathbf{X}_{is*} \subset \mathbb{R}^3$ of M test points. The implicit nature of the GPIS does not allow to directly shape the VRS. It is needed to define a large set of test points, e.g. a dense cubic volume centered on a region of interest, and then find the isosurface of value 0 on the \bar{f}_* associated with the inferred points in \mathbf{X}_* . This operation is very computationally expensive depending on the size of the VRS and \mathbf{X}_* . On the other hand GPIS allows to model complex surfaces and to use not only points belonging on the surface to shape the GPR but also empty region points (i.e. $f(x) \neq 0$).

As we show on Sec. V, this property is critical in defining the amount of interactions needed to describe the occluded VRS. For the GPIS we choose the same rbf covariance function as in Eq. 5.

Hyper-parameters were empirically chosen based on a set of experiments made on a $1m^3$ area. Having a covariance function that maps the uncertainty on input data similarly for the bi-dimensional case and the three-dimensional case is a fundamental assumption for our analysis as we show later in section V-B.1.

¹ Axis are described considering the frame represented in Fig.1

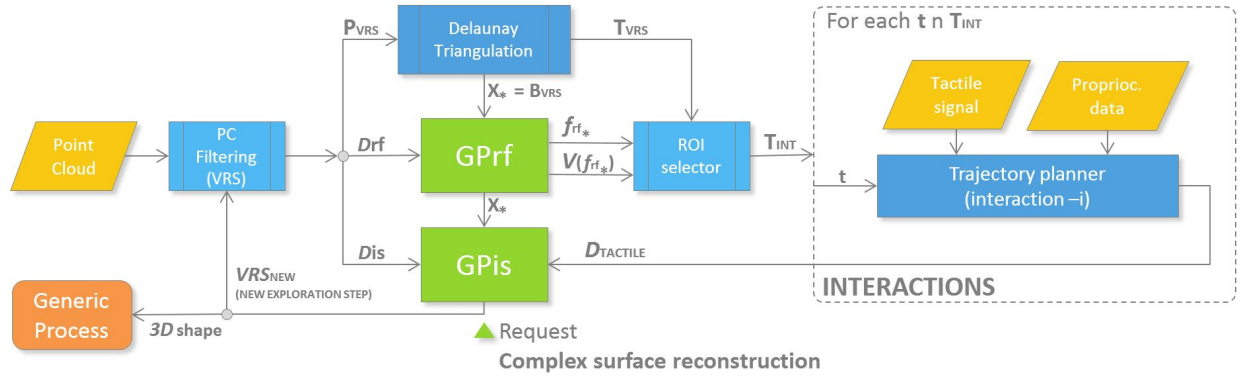


Fig. 3: Probabilistic framework processes flow

IV. METHODOLOGY

In our study we represent the environment assuming a limited number of interactions in a constrained time window considering a VRS containing multiple objects. This applies to all those robotics problems in which a robot needs to explore an environment in an online process that extends within limited amount of time (as those listed in Sec. II). In this section we describe the system and guide the reader through the framework process flow showed in Fig.3.

A. Strategy for modeling and inference

We initially model the VRS surface with a 2.5D function $f_{VRS}(\mathbf{x}) : \mathbb{R}^2 \rightarrow \mathbb{R}$ using a bi-dimensional Gaussian Process Regression defined in Sec. III-A. Generalization provided by equations 3 and 4 are generally used to obtain a DEM of the VRS and the level of confidence in the data for each point. The choice of \mathbf{X}_* is therefore crucial for understanding properties of the environment. A naive approach could create a dense grid of bi-dimensional points on the whole surface (so that every single hole in the map is somehow inferred by the GPR). This creates a large set \mathbf{X}_* that leads to a large covariance matrix even for the bi dimensional case. Instead of creating the grid we query only small empty regions analyzing the Voronoi diagram [29] of P_{VRS} defined in III-A. We run a fast 2.5D Delaunay triangulation² on the set P_{VRS} and fill \mathbf{X}_* with the xy-coordinates of the barycenters (\mathbf{B}_{VRS}) of all the computed Delaunay triangles (\mathbf{T}_{VRS}). Elements in \mathbf{X}_* represent coordinates of empty spaces and could easily be reduced in number, if needed, by putting a constraint on the area of the triangles to be analyzed (larger areas mean larger empty regions). We then use Eq. 4 to get the variance on the test points, i.e. the confidence seen as complexity of the shape or lack of information. Computing $V[f_{rf_*}]$ we can select the points in \mathbf{B}_{VRS} carrying highest uncertainty (ROI selector box in Fig.3) and therefore also the vertices of the corresponding triangles that we denote as $\mathbf{T}_{INT} = [\mathbf{b}_1, \mathbf{tr}_1], [\mathbf{b}_2, \mathbf{tr}_2] \dots [\mathbf{b}_L, \mathbf{tr}_L]$ with $L \ll N$. This step is crucial for the algorithm efficiency, allowing to obtain several regions of interest in a fast way without

actually inferring a Gaussian Process Implicit Surface on the whole cube containing the VRS.

Each \mathbf{b}_i is considered as a target position point for the trajectory planner for the arm. We define the approach vector for each target point (i.e. each interaction) by computing the normal vector to the plane defined by the triangle vertices constrained with direction going inside the surface (each $\mathbf{t} \in \mathbf{T}_{INT}$).

We define an implicit surface by the support points of a function $\Psi_{VRS}(x) : \mathbb{R}^3 \rightarrow \mathbb{R}$ using a Gaussian Process implicit surface Regression defined in Sec. III-B. We train the model using all the 3D points in P_{VRS} labeled as 0 with the addition of a smaller³ set of exterior points labeled according to Eq. 6. Artificial points above and below surface are created by increasing and decreasing the z-coordinates of copies of uniformly randomly selected points in P_{VRS} respectively.

B. Surface exploration

The tactile exploration task starts by sending trajectories to the robotic manipulator equipped with tactile sensors. From each sensor we obtain a temporal signal as a sequence of 3D points (sensor positions w.r.t. the world frame) along with their contact forces expressed as 3D vectors. We define a new training set $D_{tactile} = \{\mathbf{x}_i, y_i\}_{i=1}^N$ containing sampled 3D sensor positions labeled as 0 if there is contact (estimated from the module of the contact force vector) or 1 (i.e. above the surface) if there is no contact. In case of contact we further add below-the-surface samples as virtually generated 3D points placed a few millimeters from the contact positions along the direction of the contact force (see Fig.6). We train again the GPIS model alone adding $D_{tactile}$ to the initial training set. As we show in the experiment section on-surface points (contact points) and off-surface points are equally important when defining a surface shape. When the arm approaches the surface we start adding above-surface points to the GPIS model that will "push down" the uncertainty and redefine the implicit surface with a "clay-like" behavior. Inside an exploration step the framework updates the internal GPIS many times and queries only the GRF. This is done

²FADE25D C++ Library

³We used $1/5N$ points above and $1/5N$ points below the surface.

so that an external concurrent process can require the last world representation available at any time and interrupt the exploration step if needed (e.g. an external process queries a small portion of space using the GPIS and identifies that the analyzed occlusion contains an object so no further interactions are required). When all the trajectories have been used the exploration step is completed. The inferred mean of the Gaussian random field \bar{f}_{VRS^*} along with its variance can help reducing the dimension of the test set of the GPIS in case a full 3D model of the VRS is needed. Alg. 1 shows one way to do it. It creates a test set as a grid of 2D points on the space covered by the VRS and then computes the mean function along with the variance from the GRF. It then generates 3D points for the GPIS test set using the means as xy-coordinates and the variances as confidence intervals where to span the z-coordinates. Such simple approach can help generating a test set with dimensionality considerably smaller than a dense cube of 3D points. The new VRS obtained inferring the GPIS on the new test set can be used to close the modeling loop and trigger a new iteration of the algorithm for better shape accuracy.

Algorithm 1 Generate a test set for a GPIS from an inferred GRF

```

1: procedure GENERATESUBSET
2:    $X_* \leftarrow 2Dgrid(size)$ 
3:    $f_{rf_*} \leftarrow gpmean(X_*)$  ▷ from the GRF
4:    $V_{rf_*} \leftarrow gpvar(X_*)$  ▷ return the diagonal
5:   for all  $\mathbf{x}$  in  $X_*$  do
6:      $y \leftarrow f_{rf}(\mathbf{x}) - (m * V_{rf_*}(\mathbf{x}) + \tau_v)$  ▷ lower
       height,  $m$  and  $\tau_v$  constants
7:     while  $y < f_{rf}(x) + (m * V_{rf_*}(\mathbf{x}) + \tau_v)$  do
8:        $X_{is_*} \leftarrow addNewPointToSet(\mathbf{x}, y)$ 
9:        $y \leftarrow y + \Delta_y$  ▷  $\Delta_y$  incremental constant
10:    end while
11:  end for
12:  return  $X_{is_*}$ 
13: end procedure

```

V. EXPERIMENTAL EVALUATION

In the following we describe the experimental setup showed in Fig. 1 and the experimental scenarios.

A. Experimental setup

The point cloud is obtained from a PrimeSense 3D camera placed 60 cm above a table oriented to form a 35° angle with the table plane. The table surface can be configured to contain holes, reflective surfaces or soft surfaces in order to recreate different scenarios. The tactile sensory system is composed of a Kinova Jaco2⁴ robotic arm (6 dof) with a 3 fingered Kinova KG-3 gripper equipped with 3D OptoForce force sensors⁵. The tactile sensors can detect slipping and

shear forces with high frequency. We use the sensor output to obtain 6D force-position signals which is used for generating the tactile training dataset. Proprioceptive data are less affected by noise with respect to vision sensor data. We generate contact and non-contact points after each physical interaction using the sensor orientation w.r.t. the world frame and the output contact force. We generate above-the-surface 3D points as square grids of 16 points placed along the downsampled fingertip trajectories (i.e. sensor position) oriented accordingly. The size of each grid is 8mm×8mm as the spherical sensor dimension and consistent with the VRS point cloud density. We use the contact force direction to orient the on-surface points grid to be orthogonal to the surface normal at the contact position. Below-the-surface points are virtually generated only in case of contact, translating a copy of the grid of on-surface points along the surface normal (see Fig. 6).

The arm follows the approach vector on each target triangle until contact, until the arm reaches its work space limit (i.e. VRS border) or until it diverges too much from the target (e.g. terrain holes).

Hybrid position-force control [30] is used in proximity of the target point to allow small displacement along the plane orthogonal to the approach vector while imposing a minimum contact force along the approach.

B. Scenarios

1) *Reflective surfaces*: Reflective surfaces as metal plates or water pots generate ambiguity in the point cloud representation of the environment due to photometric effects as shown in the initial VRS in Fig. 4. In the first scenario we use the presented active perception algorithm to identify and model the difference between the two incomplete point cloud regions in an online, fully autonomous fashion. We repeated the experiment using 3 different reflective shapes and holes (reflective shapes RS 1,2 and 3 in Fig. 4) on a single exploration step. The first row of Fig. 4 shows the selected regions of interest on the point cloud with the corresponding Delaunay triangles laying on the areas of high uncertainty (estimated from the GRF imposing a minimum triangle area) and approach vectors. Each triangle selection triggers a physical interaction (visible in the second row of each column). After each action we train the GPIS with the new tactile training previously mentioned. The last row shows the isosurface of value 0 representing the implicit surface modeled by the GPIS after each interaction. We invite the reader to note how the last two interactions allow to shape the hole but do not reduce the representation uncertainty in that region. This is because we fed the model with above-the-surface points (i.e. non contact points since the arm could not reach the bottom of the hole) that only helped identify areas where the implicit surface could not be.

After the exploration step we used a simple threshold-based binary terrain classifier to automatically label the holes in the analyzed areas. Example of more advanced terrain classification can be seen in [31] and [32]. Results using 3

⁴Kinova website: <http://www.kinovarobotics.com/>

⁵Optoforce website: <http://optoforce.com/>

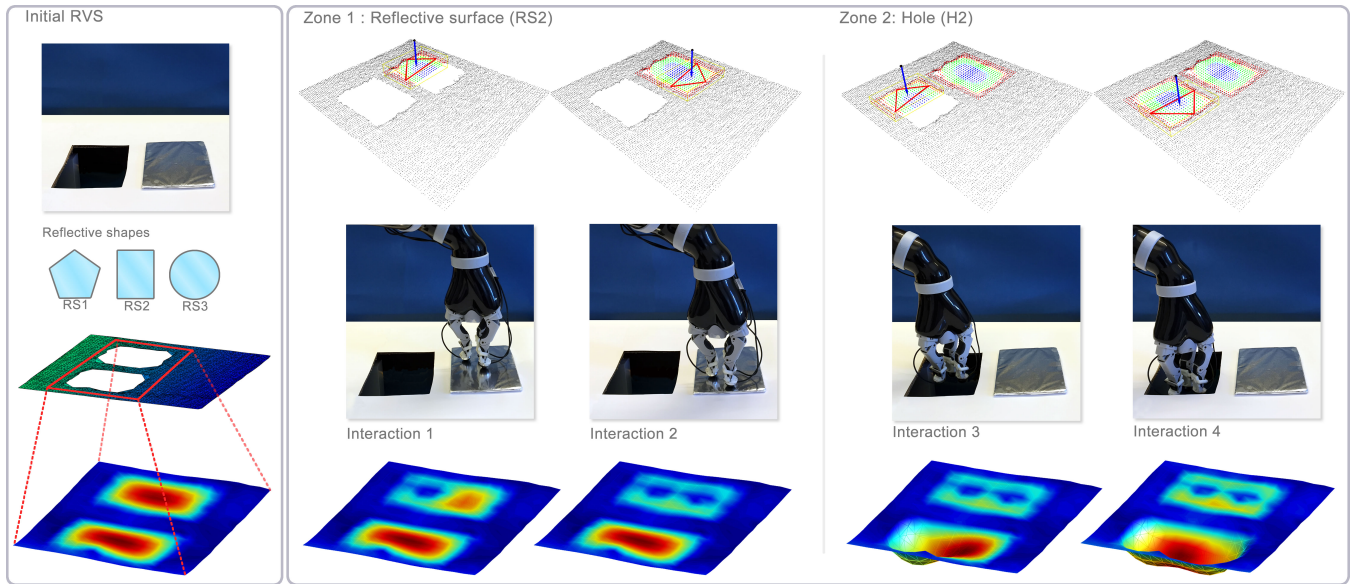


Fig. 4: Exp. setup 1: Different reflective objects are placed next to holes of similar shape which generates ambiguity on the VRS point cloud. The algorithm identifies regions of high uncertainty and starts poking the surface on different locations. After 4 interaction, the 3D model allows to clearly identify the two elements. Blue color in the first row indicates high variance in the GRF model queried in proximity of the select Delaunay triangles. Red color on the third row indicates high uncertainty in the GPIS model.

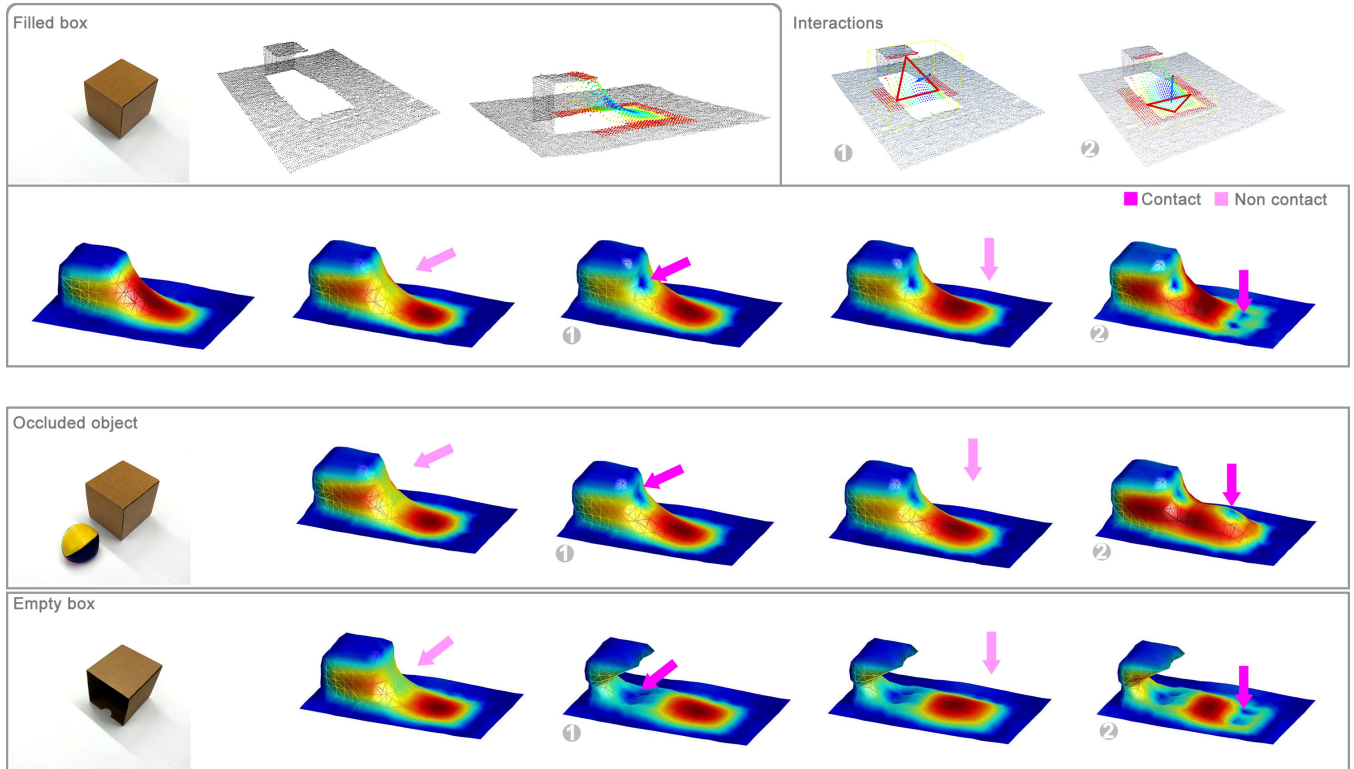


Fig. 5: Exp. setup 2: An object generates an occluded area on the VRS (first row). The algorithm analyses three different situations where the occluded area is a flat terrain (second row), hides a different object (third row) or has complex shapes (fourth row). After 4 interaction, the 3D model allows to describe the occluded area. Colormaps for points (first row) and shapes are chosen as described in Fig.4. In the last scenario the model identifies the empty area inside the box but creates an artifact (two missing faces) due to a lack in lateral interactions.

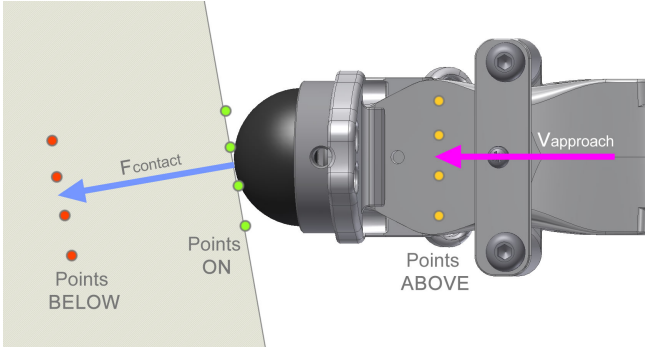


Fig. 6: Top view of a finger with positions of on and off surface points after contact.

different reflective surfaces and hole shapes are shown on Table I on a $0.7\text{m} \times 0.6\text{m} \times 0.6\text{m}$ VRS.

Scen.	Shape in the first region	Shape in the second region	n° interact.	T_h Classifier [hole/flat/object]
1	RS1	H1	8	(?,?) \rightarrow (flat,hole)
2	RS2	H2	4	(?,?) \rightarrow (flat,hole)
3	RS3	H3	6	(?,?) \rightarrow (flat,hole)
4	RS1	RS2	6	(?,?) \rightarrow (flat,flat)
5	H1	H2	7	(?,?) \rightarrow (hole,hole)

TABLE I: Different combinations of reflective surface (RS) and hole (H) shapes placed in two different regions. A simple threshold classifier T_h labels the presence of holes using two average heights centered on the query regions as shown in Fig.4.

The exploration steps, including elaboration time⁶, planning and physical interactions lasted between 3-5 minutes for the above mentioned scenarios. The dimension of P_{VRS} varied between $N = 5000$ and $N = 21000$ points depending on the dimension of the occlusions and manipulability constraint. B_{VRS} contained between 12 and 21 baricenter points. Dimension of T_{INT} (L , resulting after reduction based on GRF variance) was between 4 and 8 (n° interact.).

2) *Occluded areas:* In the second experiment (Fig. 5) we demonstrate how the algorithm can reconstruct occluded areas and how it can extract environmental properties which are not visible in a simple DEM. Similarly to the previous scenarios we analyse (now independently) elements placed inside the VRS that have the same point cloud representation as shown in Fig. 5. The first element is a full cubic box that generates a large occluded area on the initial VRS point cloud. The second element is an empty cubic box that hide its open face from the camera. The empty area inside the cube cannot be represented by a DEM (GRF) that would only consider the height of the upper side on the box. The third object is the same full cubic box that hides a third different object (a soft ball) placed a few centimeter behind it. By limiting the area of the Delaunay triangles we force the algorithm to only have 2 interactions on the exploration step for each scenario. The first row of Fig. 5 shows the the point cloud representation of the full

box on the VRS together with its uncertainty distribution generated by the GRF and the process of action selection with the selected Delaunay triangles. Second row shows the evolution of the internal representation (computed offline) during each interactions. It is possible to notice how the implicit surface changes while the arm approach the target. Second row shows the evolution of the GPIS model for the second scenario. The occluded object becomes visible only during the second interactions after a physical contact. Last row shows that the GPIS can generalize information more complex than the ones embedded on a DEM. The empty cube shaped is revealed by the first interaction that do not add any on-surface point. The box is in fact carved by the off-surface points. The two triangles carrying higher uncertainty selected during the exploration step do not generate any lateral interaction with the box. This results in an artifact in the internal representation (last row of Fig.5) where the two lateral faces of the box disappear. Such situation can be avoided increasing the number of interactions for each exploration step. Similarly to the previous experiment we use a simple threshold classifier on the occluded area to identify the presence of objects as shown in Table II.

Scen.	Scenario description	n° interact.	T_h Classifier [object/flat/hole]
1	Full box	4	(obj,?) \rightarrow (obj,flat)
2	Box with occluded object	4	(obj,?) \rightarrow (obj,obj)
3	Empty box	4	(obj,?) \rightarrow (obj,flat)

TABLE II: Detection of occluded objects. A simple threshold classifier T_h labels the presence of objects using two average heights centered on the box and on the occluded area behind the box respectively.

VI. CONCLUSIONS

We presented an efficient probabilistic framework for building a 3D model of a surface containing different occluded areas, objects and reflective surfaces⁷. The algorithm uses Delaunay triangulation and Gaussian Random Fields to quickly identify areas poorly described by the visual sensory system avoiding the computational cost of Gaussian Implicit Surfaces. The system generates subsequent target positions and orientations for a trajectory planner that brings a robotic arm equipped with tactile force sensor to touch the uncertain regions of the local map. On-surface and off-surface points generated during each physical interaction of the arm are used to update a Gaussian process implicit surface that keeps an internal complex representation of the environment. We did real experiments to show how very few interactions can unveil fundamental information hidden in the environment. We also showed how off-surface points alone (that are generated in case of non contact trajectories) can help to model simple terrain shapes. The algorithm can be used in an online process as opposed to other methods [3] and can be iterated to increase the quality of the 3D model. A limitation of the framework appears when the terrain complexity

⁶Using PCL, Eigen, ROS, Kinova SDK

⁷Video of an experiment available at: <https://youtu.be/0-UIFRQT0JI>

increases or when the covariance functions used for the GPIS and the GRF differ considerably. In such cases tactile interactions (that are planned using the GRF model) cannot bring enough information to the GPIS, resulting in wrong surface representations. A second weakness arises when the arm modifies too much the surface under analysis during the physical interactions and the internal representation of the environment diverges from the real world. In future work we plan to study these problems by segmenting objects in the environment and incorporating relative translations into the model. Variance values inside the triangles can help to generate sliding-on-surface acquisitions to obtain more tactile data from each interaction and embed additional surface properties. We also plan to test the algorithm on the robot shown in Fig. 2 and on a PR2.

ACKNOWLEDGMENTS

The authors gratefully acknowledge funding under the European Union's seventh framework program (FP7), under grant agreements FP7-ICT-609763 TRADR.

REFERENCES

- [1] J. Bohg, M. Johnson-Roberson, B. Leon, J. Felip, X. G. Martínez, N. Bergström, D. Kragic, and A. Morales, "Mind the gap - robotic grasping under incomplete observation," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 686–693.
- [2] M. Meier, M. Schopfer, R. Haschke, and H. Ritter, "A probabilistic approach to tactile shape reconstruction," *Robotics, IEEE Transactions on*, vol. 27, no. 3, pp. 630–635, June 2011.
- [3] M. Bjorkman, Y. Bekiroglu, V. Hogman, and D. Kragic, "Enhancing visual perception of shape through tactile glances," pp. 3180–3186, Nov 2013.
- [4] H. B. Helbig and M. O. Ernst, "Optimal integration of shape information from vision and touch," *Experimental Brain Research*, vol. 179, no. 4, pp. 595–606, 2007.
- [5] M. O. Ernst and M. S. Banks, "Humans integrate visual and haptic information in a statistically optimal fashion," *Nature*, vol. 415, no. 6870, pp. 429–433, 2002.
- [6] I. Kruijff-Korbayová, F. Colas, M. Gianni, F. Pirri, J. de Greeff, K. Hindriks, M. Neerinx, P. Ögren, T. Svoboda, and R. Worst, "Tradr project: Long-term human-robot teaming for robot assisted disaster response," *KI - Künstliche Intelligenz*, vol. 29, no. 2, pp. 193–201, 2015. [Online]. Available: <http://dx.doi.org/10.1007/s13218-015-0352-5>
- [7] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006. [Online]. Available: <http://www.gaussianprocess.org/gpml/chapters/>
- [8] S. Vasudevan, "Data fusion with gaussian processes," *Robotics and Autonomous Systems*, vol. 60, no. 12, pp. 1528 – 1544, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0921889012001388>
- [9] S. O'Callaghan, F. Ramos, and H. Durrant-Whyte, "Contextual occupancy maps using gaussian processes," in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, May 2009, pp. 1054–1060.
- [10] R. J. Peckham and G. Jordan, *Digital Terrain Modelling: Development and Applications in a Policy Support Environment*. Springer Science & Business Media, 2007.
- [11] S. Vasudevan, F. Ramos, E. Nettleton, H. Durrant-Whyte, and A. Blair, "Gaussian process modeling of large scale terrain," in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, May 2009, pp. 1047–1053.
- [12] G. Turk and J. F. O'brien, "Variational implicit surfaces," 1999.
- [13] J. F. Blinn, "A generalization of algebraic surface drawing," *ACM Trans. Graph.*, vol. 1, no. 3, pp. 235–256, Jul. 1982. [Online]. Available: <http://doi.acm.org/10.1145/357306.357310>
- [14] J. Solem and A. Heyden, "Reconstructing open surfaces from unorganized data points," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2, June 2004, pp. II–653–II–660 Vol.2.
- [15] F. Steinke, B. Schlkopf, and V. Blanz, "Support vector machines for 3d shape processing," *Computer Graphics Forum*, vol. 24, no. 3, pp. 285–294, 2005. [Online]. Available: <http://dx.doi.org/10.1111/j.1467-8659.2005.00853.x>
- [16] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H.-P. Seidel, "Multi-level partition of unity implicits," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 463–470, Jul. 2003. [Online]. Available: <http://doi.acm.org/10.1145/882262.882293>
- [17] O. Williams and A. Fitzgibbon, "Gaussian process implicit surfaces," *Gaussian Proc. in Practice*, 2007.
- [18] M. A. El-Beltagy and W. A. Wright, "Gaussian processes for model fusion," in *Artificial Neural Networks ICANN 2001*. Springer, 2001, pp. 376–383.
- [19] M. Gerardo-Castro, T. Peynot, and F. Ramos, "Laser-radar data fusion with gaussian process implicit surfaces," in *Field and Service Robotics*, ser. Springer Tracts in Advanced Robotics, L. Mejias, P. Corke, and J. Roberts, Eds. Springer International Publishing, 2015, vol. 105, pp. 289–302.
- [20] S. Dragiev, M. Toussaint, and M. Gienger, "Gaussian process implicit surfaces for shape estimation and grasping," pp. 2845–2850, May 2011.
- [21] A. Melkumyan and F. Ramos, "A sparse covariance function for exact gaussian process inference in large datasets," in *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, ser. IJCAI'09. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2009, pp. 1936–1942. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1661445.1661755>
- [22] N. D. Lawrence, M. Seeger, and R. Herbrich, "Fast sparse gaussian process methods: The informative vector machine," in *Advances in Neural Information Processing Systems 15*. MIT Press, 2003, pp. 609–616.
- [23] K. Soohwan and K. Jonghyuk, "Gpmap: A unified framework for robotic mapping based on sparse gaussian processes," in *Field and Service Robotics*, ser. Springer Tracts in Advanced Robotics, L. Mejias, P. Corke, and J. Roberts, Eds. Springer International Publishing, 2015, vol. 105, pp. 319–332.
- [24] G. A. Hollinger, B. Englot, F. S. Hover, U. Mitra, and G. S. Sukhatme, "Active planning for underwater inspection and the benefit of adaptivity," *Int. J. Rob. Res.*, vol. 32, no. 1, pp. 3–18, Jan. 2013. [Online]. Available: <http://dx.doi.org/10.1177/0278364912467485>
- [25] L. Zhang and J. Trinkle, "The application of particle filtering to grasping acquisition with visual occlusion and tactile sensing," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, May 2012, pp. 3805–3812.
- [26] A. Petrovskaya, O. Khatib, S. Thrun, and A. Y. Ng, "Touch based perception for object manipulation."
- [27] K. Zimmermann, P. Zuzánek, M. Reinstein, T. Petricek, and V. Hlaváč, "Adaptive traversability of partially occluded obstacles," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, 2015.
- [28] K. Zimmermann, P. Zuzánek, M. Reinstein, and V. Hlaváč, "Adaptive traversability of unknown complex terrain with obstacles for mobile robots," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 5177–5182.
- [29] S. Fortune, "Handbook of discrete and computational geometry," J. E. Goodman and J. O'Rourke, Eds. Boca Raton, FL, USA: CRC Press, Inc., 1997, ch. Voronoi Diagrams and Delaunay Triangulations, pp. 377–388. [Online]. Available: <http://dl.acm.org/citation.cfm?id=285869.285891>
- [30] W. D. Fisher and M. S. Mjtaba, "Hybrid position/force control: a correct formulation," *The International journal of robotics research*, vol. 11, no. 4, pp. 299–311, 1992.
- [31] A. Krebs, C. Pradalier, and R. Siegwart, "Comparison of boosting based terrain classification using proprioceptive and exteroceptive data," in *Experimental Robotics*. Springer, 2009, pp. 93–102.
- [32] J.-F. Lalonde, N. Vandapel, D. F. Huber, and M. Hebert, "Natural terrain classification using three-dimensional ladar data for ground robot mobility," *Journal of field robotics*, vol. 23, no. 10, pp. 839–862, 2006.