Active Perception: Building Objects' Models using Tactile Exploration

Nawid Jamali[1], Carlo Ciliberto[2], Lorenzo Rosasco[3] and Lorenzo Natale[1]

***Abstract*— In this paper we present an efficient active learning strategy applied to the problem of tactile exploration of an object's surface. The method uses Gaussian process (GPs) classification to efficiently sample the surface of the object in order to reconstruct its shape. The proposed method iteratively samples the surface of the object, while, simultaneously constructing a probabilistic model of the object's surface. The probabilities in the model are used to guide the exploration. At each iteration, the estimate of the object's shape is used to slice the object in equally spaced intervals along the height of the object. The sampled locations are then labelled according to the interval in which their height falls. In its simple form, the data are labelled as belonging to the object and not belonging to the object: object and no-object, respectively. A GP classifier is trained to learn the object/no-object decision boundary. The next location to be sampled is selected at the classification boundary, in this way, the exploration is biased towards more informative areas. Complex features of the object's surface is captured by increasing the number of intervals as the number of sampled locations is increased. We validated our approach on six objects of different shapes using the iCub humanoid robot. Our experiments show that the method outperforms random selection and previous work based on GP regression by sampling more points on and near-the-boundary of the object.**

I. Introduction

Knowledge of an object's shape and its surface properties such as edges and surface curvature are important. This information, for example, can be used to identify the object, which in turn, can help in manipulating the object using methods such as form and force-closure [1]. Object shape can be reconstructed using depth cameras [2], [3]. Often sensing modalities such as vision – due to occlusion, illumination, background, and pose of the object relative to the camera – are limited or unreliable, hence, direct contact information is important to successfully inspect an object [4]. Sensing object properties through contact is a time consuming task, for example, constructing a point cloud of an object using contact requires sampling multiple locations in order to cover the surface of the object. Active perception can reduce the number of contacts required to perceive an object.

This research has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement No. 610967 (TACMAN).

[1]The authors are with the iCub Facility, Istituto Italiano di Tecnologia, via Morego, 30, 16163 Genova, Italy. Email: {nawid.jamali, lorenzo.natale}@iit.it

[2] Laboratory for Computational and Statistical Learning, Isitituto Italiano di Tecnologia via Morego, 30, 16163 Genova, Italy and Poggio Lab, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA, USA. Email: cciliber@gmail.com

[3]DIBRIS, Università degli Studi di Genova, Genova, Italy and Poggio Lab, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA, USA. Email: lrosasco@mit.edu

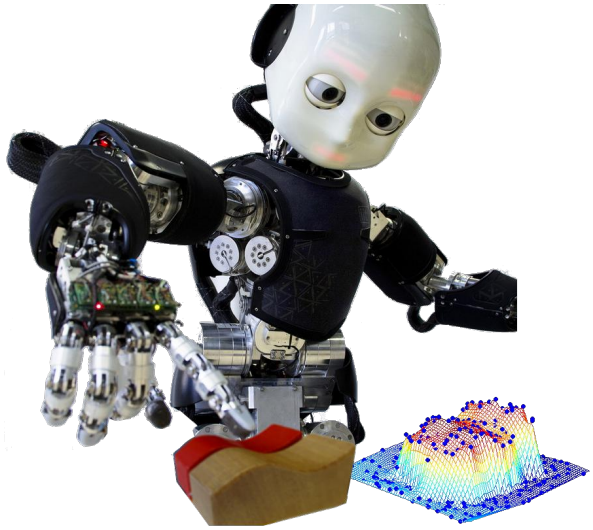


Fig. 1. The experimental setup: the iCub robot exploring an object using its index finger. The fingertip is equipped with a tactile system which has 12 taxels. The figure also shows a mesh of the surface generated from locations sampled by the robot in a grid of points with a $5\,mm \times 5\,mm$ cell size.

In this paper, we propose a method based on active perception that reduces the number of samples required to construct a three-dimensional point cloud of an object to capture its shape, i.e., its edges and surface curvature. In the proposed method, the robot, iteratively, makes contact with an object. At each iteration, the data collected hitherto is used to construct a probabilistic model of the object's surface. We use the probabilities in the model to guide the surface exploration. To this end, we use a Gaussian process (GP) classifier. Any probabilistic classifier can be used, however, an advantage of GPs is that they are non-parametric, that is, instead of fitting parameters to a model, they infer how the training data are correlated. Consequently, we are able to build reliable models of the surface with a small number of training data. The main contribution of the paper is the exploration strategy, in which, we use active probabilistic classification to determine where to sample the object such that with a smaller number of contacts it is possible to capture its shape.

The method works by labelling the sampled points as belonging to the object, and not belonging to the object: object and no-object, respectively. A simple heuristic based on the height of the contact locations is used to label the sampled points. A Gaussian process classifier is learned using the points sampled thus far. The next location to sample is selected at the classification boundary. This ensures that the areas that do not belong to the object are not unnecessarily sampled. The same principle is applied to distinguish finer

features of the object by increasing the number of classes. That is, classifying between points of different height on the object. When the number of classes is increased, all of the points already collected still contribute to the analysis, thereby, reducing the number of locations required to sample when the complexity of the model is increased. We applied the approach to six objects and we show that the proposed method performs better than random selection and previous approaches based on GP regression [5].

II. Background

The problem of object exploration using contact is challenging as it requires physical interaction with the object. Literature on object exploration can be divided into two categories: object property exploration and object shape reconstruction. The former focuses on using contact, namely, tactile sensors, to differentiate objects based on texture [6], [7], [8], encode surface roughness [9] and surface curvature [10]. The latter focuses on building a three-dimensional model of the surface of the object. Object shape, among other methods, can be discovered by contour following [11], [12], and constructing a 3D point cloud [5].

An earlier work on using tactile driven object exploration is presented by Natale and Torres-Jara [13]. The authors implemented a control strategy that relies on tactile feedback to grasp objects and demonstrate that the information originated by the interaction carries implicit information about their shape and can be useful for learning. Recently, Martinez-Hernandez et al. [11] proposed an approach in which tactile sensors are used to follow the edges of an object. They used a tapping motion to sample the surface. At each iteration, the tactile sensor data are processed to discriminate edges and flat regions on the object. The relative angle between the finger and the object's edge is used to follow the edges of the object.

The surface of an object can also be sampled by tracing its surface. Liu et al. [12] propose a method to follow the surface of the object based on a controller proposed by Nguyen and Perdereau [14]. They use surface normal, contact normal force and the local torque to guide the fingertip over the surface from a starting location to a desired location. Montana [15] derived a set of equations describing the kinematics of contact between two rigid bodies. The author used these equations to determine the curvature of a surface at the point of contact, which was subsequently used to trace the surface. Methods based on surface contour following reduce the number of contacts required to explore a surface, however, they can benefit from active exploration strategies to quickly discover object properties of interest.

Active exploration strategies have been used to efficiently classify objects. Fishel and Leob [6] propose an active exploration strategy that they call Bayesian exploration. The algorithm selects the optimal exploratory behaviour that distinguishes a texture from a set of plausible candidates. They show that, by actively selecting an exploratory behaviour, they can increase the accuracy of the classifier with a small number of samples. Martins et al. [7] propose an active exploration approach using Bayesian models. Their approach consists of two Bayesian models running concurrently, one determines, based on the object's surface texture, the probability of the object in contact with the fingertip. The other model determines the next location to be sampled with the aim of following the boundary between two textures

Dragiev et al. [16] use Gaussian processes (GPs) to fuse tactile and depth-camera data to model an implicit representation of the surface of an object. In the model, a point on the surface is represented by zero; a point inside the object and a point outside the object are represented by negative and positive numbers, respectively. The surface representation is aimed at guiding a robotic manipulator towards the object surface. It is not used to actively construct an explicit representation of the surface of the object.

Björkman et al. [17] use GPs to construct the model of an object using vision. The model is then refined by making contact with object at the surface points where the GP uncertainty is high. Recently, Yi et al. [5] applied GP regression to the problem of object exploration. They model the object surface by fitting a GP model to the acquired point cloud. The uncertainty given by the GP is used to determine the next location to explore. They show that this approach reconstructs more accurately a 1-D surface with fewer points than a random approach. They also applied the method to reconstruct the surface of two everyday objects. This work is closely related to the work presented in this paper. We use the same problem formulation, however, we differ in the way we apply the GP to guide the object exploration. In particular, we develop an active strategy that uses GP classification to bias the exploration towards the object. Similar to GP regression, the parameters of a GP classifier are inferred from the data.

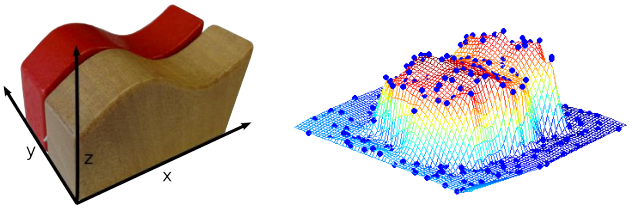
III. Methodology

We propose a method, based on GPs, to efficiently sample the surface of an object such that with a smaller number of contact locations we can reconstruct the shape of the object, capturing its edges and surface curvature. In this section, we start with formalizing the problem, which is followed by a brief description of GPs and how we apply it to the problem at hand. We end the section with a detailed description of the proposed method.

A. Problem Formulation

Given a finite number of observed contact locations and the associated object heights, the problem is divided in two parts: 1) fit the best object shape to the observed data, and 2) identify the next contact location to explore that captures more information about the object's shape, edges and surface.

We define a contact location by the Cartesian x, y and z coordinates of the robot's fingertip, at the finger-object interface, in the robot's frame of reference. The object is assumed to be positioned on a horizontal table, resting on a stable face. We also assume that the position of the object is fixed during the exploration. As illustrated in Fig. 2(a), the x and the y coordinates of the object are parallel to the table with the z coordinate pointing out of the table. The robot



(a) (b)

Fig. 2. An object (2(a)) and the corresponding point cloud (2(b)) collected at various locations in the x-y plane. The object is sampled in a grid of points with a $5\,mm \times 5\,mm$ cell size.

samples the x-y plane to discover the object's surface. In other words, the robot selects a location, (x, y), and samples the z coordinate of the location, which corresponds to the surface of the object. Figure 2(b) shows a point cloud of the object's surface generated by sampling the object in a gird of points with $5\,mm \times 5\,mm$ cell size.

We formulate the problem as a standard learning problem where we learn the object shape, $f(x, y) = z$, which is used to find the next sampling location, (x, y), such that the shape estimation error decreases faster.

B. Gaussian Processes

Here we briefly review the main ideas related to Gaussian processes (GPs). We refer the reader to [18] for a thorough introduction. Gaussian processes are a supervised learning approach whose goal is to learn a functional relation $f : \mathcal{X} \rightarrow \mathcal{Y}$ between an input set $\mathcal{X}$ and an output set $\mathcal{Y}$, given only a finite set of observations[1]. In this work we consider $\mathcal{X} = \mathbb{R}^2$, the first two components of the Cartesian coordinates of a contact location. The output – as described next – depends on whether we are using regression or classification.

1) Regression: A GP models a set of, n, observations $\{x_i, y_i\}_{i=1}^n$ as a multivariate normal distribution $\mathcal{N}(\bar{f}(x), v(x))$ in which the mean and variance of $f(x)$ can be computed in closed form as:

$$\bar{f}(x) = \mathbf{k}_x^\top (\mathbf{K} + \lambda I)^{-1} \mathbf{y} \tag{1}$$

$$v(x) = k(x, x) - \mathbf{k}_x^\top (\mathbf{K} + \lambda I)^{-1} \mathbf{k}_x \tag{2}$$

where $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is a so-called "covariance" function and $\mathbf{k}_x \in \mathbb{R}^n$ the vector with i^{th} entry $k(x, x_i)$ and $y \in \mathbb{R}^n$ the collection of training outputs y_i. A wide range of possible choices of covariance functions k have been proposed in the literature (see [18] for a partial list). For the experimental analysis we performed in this work, we chose the radial basis function (RBF):

$$k(x, x') = \sigma^2 exp\left(-\frac{(x - x')^2}{2l^2}\right) + \lambda \delta_{x=x'} \tag{3}$$

which allows modeling complex functions and is a well-established covariance function for GPs. In Eq. (3), $\delta_{x=x'}$ is the function constantly zero except when x and x' are identical.

Interestingly, by modeling the desired function as a Gaussian process, we have access to both the mean value $\mathbb{E}(f(x))$ and its variance $Var(f(x))$. The first term reports the value minimizing the mean squared error $\mathbb{E}((y - f(x))^2)$ and therefore provides us with an estimate of the expected values of $f(x)$. The second term informs us about how "certain" our model is about such prediction. Indeed, for small values of $v(x)$, the normal distribution $\mathcal{N}(\bar{f}(x), v(x))$ will be strongly concentrated around the expected value $\bar{f}(x)$, while for large values the prediction $\bar{f}(x)$ will be much less reliable. This observation suggests a possible direction for active learning, where the learning system requires novel points to be sampled from regions where $v(x)$ is larger in order to reduce the uncertainties in those regions at the next learning step. In the method proposed in this paper, the output $\mathcal{Y}$ is a real number corresponding to the height of the object at the sampled location.

2) Classification: In classification settings the output set $\mathcal{Y}$ is a collection of two or more labels (e.g. $\mathcal{Y} = \{1, \ldots, C\}$ $C \in \mathbb{N}$) and the goal is to assign the correct label to every example $x \in \mathcal{X}$. Several modeling approaches have been proposed for classification with GPs [18]. Here we consider the simple strategy where each class $c = 1, \ldots, C$ is mapped to vector $v_c \in \mathbb{R}^C$ with all entries equal to -1 but the c^{th} entry, which is set equal to 1. The learning problem is then cast as a multi-variate regression problem, namely the goal is to learn a function $\bar{f} : \mathcal{X} \rightarrow \mathbb{R}^C$ from observations $\{(x_i, v_{c_i})\}_{i=1}^n$ as in (1). Then, given a new point x, its class is chosen as

$$\hat{c}(x) = \underset{c=1,\ldots,C}{\text{argmax}}\ \bar{f}(x)_c. \tag{4}$$

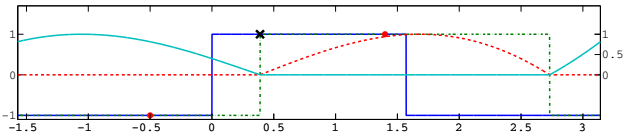
Indeed it can be proved that under suitable assumptions [19], $\bar{f}(x)_c$ converges to the probability $p(y = c|x)$ as the number of training samples tends to infinity $n \rightarrow +\infty$. Therefore the classification rule at (4) converges to the ideal Bayes' rule $c^*(x) = \text{argmax}_c p(y = c|x)$.

3) Model Selection: The parameters σ, l and λ in (3) are typically chosen by model selection [18]. For the experimental analysis in this work we adopted an hold out cross-validation strategy by separating the training set in two subsets, respectively used for training and validation. The parameters leading to the best validation error on average across multiple trials were then selected and a final model was trained on the joint train-validation dataset. This procedure was automatically performed by the GURLS machine learning library [20] in MATLAB.

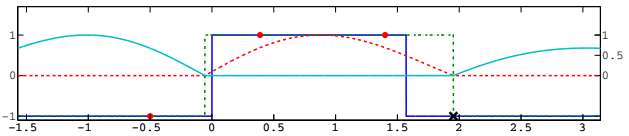
C. Next Action Selection

We will use a simple two-dimensional example to illustrate how the algorithm selects the next location to be sampled. Let's imagine we have a square shaped object on a table. As illustrated in Fig. 3, the solid blue line represents the object. Initially the robot samples two random locations. These points are used as the input of a GP classifier to form the initial hypothesis. The dashed green line represents the

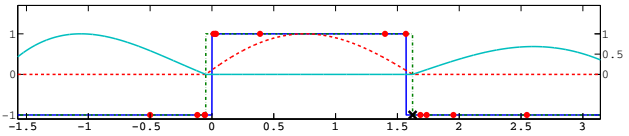
[1]Note, following the standard notation for machine learning, in this section we identify input vector with x and output data with y. It is not to be confused with x and y used earlier to denote the first two components of the Cartesian coordinates of a contact location.



(a) Two points



(b) Three points



(c) Fourteen points

Fig. 3. This figures illustrates the proposed strategy using a simple example. The solid blue line represents the object, the dashed green line represents the hypothesis of the classifier. The right vertical axis plots the probability of the hypothesis, where the red dashed line is the probability of a point belonging to the object and the cyan line for no-object. The points sampled thus far are marked by red circles, the black cross represents the next location to be sampled: a) two randomly selected points to help form the initial hypothesis, the black cross shows the next point the algorithm wants to sample to improve the hypothesis; b) shows the improved hypothesis after the new location is sampled; c) shows the hypothesis after fourteen samples. We notice most of the sampling points are near or on the object.

hypothesis of the classifier. The points sampled thus far are marked by red circles, the black cross shows the next point the robot wants to sample to improve its hypothesis.

At each iteration the algorithm updates its hypothesis on the object-table boundary by sampling a point near the boundary. Figure 3(a) shows the initial hypothesis formed using the two randomly selected locations. The black cross marks the next point the robot wants to sample. Figure 3(b) shows the improved hypothesis after the new location is sampled. By sampling locations near the hypothesis boundary, the robot quickly converges its hypothesis to the ground truth. Figure 3(c) shows the hypothesis after fourteen locations are sampled.

In this manner the robot is biased towards sampling points on or near the object boundary, reducing the number points needed to uncover the shape of the object. Once the object-table boundary is defined, the same principle is applied to distinguish finer features of the object by increasing the number of classes. That is, classifying between points of different height on the object. When the number of classes is increased, all of the points already collected still contribute to the analysis, thereby, reducing the number of points required to sample as the complexity of the model is increased.

Algorithm 1 summarises the proposed algorithm. Given the workspace, W, a set, S, of locations already sampled, it computes the next location, (x_{t+1}, y_{t+1}), to sample. As shown in line 5, the algorithm can be divided into two main parts: exploitation – the Gaussian process classification, and exploration – the Gaussian process regression. The exploitation component, which is the main contribution of

Algorithm 1: Next Action Selection

input : $S = \{(x_1, y_1, z_1)...(x_t, y_t, z_t)\}$
$W(x_{min}, x_{max}, y_{min}, y_{max})$
output: (x_{t+1}, y_{t+1})

0 $\alpha \leftarrow 0.3$
1 $n \leftarrow$ number of classes (Section III-C.1)
2 $L \leftarrow$ class labels (Section III-C.2)
3 $P_c^{x,y} \leftarrow GP_{classification}$ (Section III-C.3)
4 $P_r^{x,y} \leftarrow GP_{regression}$ (Section III-C.4)
5 $(x_{t+1}, y_{t+1}) \leftarrow \text{argmin}\left((1-\alpha) \times P_c^{x,y} + \alpha \times P_r^{x,y}\right)$

this paper, encourages the robot to sample the object. The exploration component ensures that the robot is not trapped in local minima. The exploitation versus exploration can be controlled by adjusting the parameter α. At each time step, t, we perform the following steps (which correspond to the line numbers in Algorithm 1):

1) Class Band: The first step in the algorithm is to determine the number of class bands. We use a simple strategy to increase the number of classes during classification. For the first twenty samples the number of classes is restricted to two. After which, it is increased by two every ten samples.

2) Labelling: Learning a GP classifier requires labelled data. We use a heuristic based on the value of the z-coordinate of the locations already sampled. The heuristic divides the interval between the maximum and the minimum z values into n equally spaced intervals. The number of intervals or class-bands is determined by the method described earlier. The label of a contact location is determined by the band in which the value of its z coordinate falls.

3) Exploitation: Once the data are labelled we learn a model of the surface using the GP classifier described in section III-B.2. Then, we query the learned classifier. The query set is comprised of m locations, (x, y), evenly distributed in the workspace. This produces a probability distribution for each class over the entire workspace. At each location, we sum the probability of all classes. This gives a single probability distribution for the workspace.

4) Exploration: We also learn a model of the surface of the object using GP regression. The uncertainty in the GP regression indicates areas in the surface model that are unexplored or that are affected by larger noise. This term is used to add an exploration component to avoid the robot from being trapped in local minima. After learning the GP regression model, we use the same query vector from the GP classification to evaluate a probability distribution for the workspace.

5) Next Location: A unique value of the probability distribution $P(x, y)$ over the workspace is computed as the weighted sum of the individual probability values from the two GP models. The next sampling location (x_t+1, y_t+1) is then computed as the value of (x, y) that minimizes $P(x, y)$. In other words, we select a location where the model has the lowest confidence in its prediction. The minimization is performed numerically by sampling the workspace in a

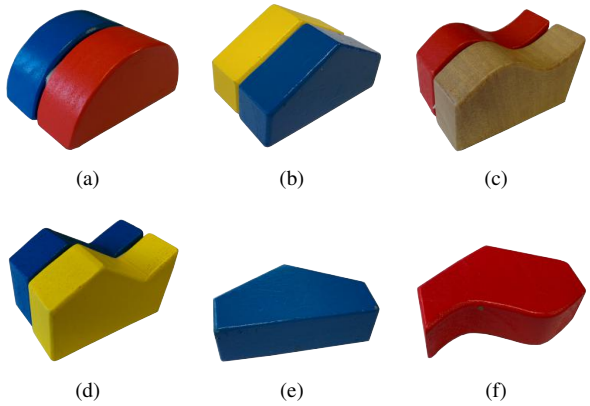


Fig. 4. Objects used for object-surface exploration and reconstruction.

uniform grid of 80×80 points.

IV. Experimental Setup

The setup to test the proposed method consists of the iCub robot [21] and six objects (Fig. 1). An object is placed on a table in the robot's workspace. The robot explores the object using its index finger by making a contact with the object. A contact event is detected by the tactile sensors in the robot's finger. In this section we will describe the objects and describe how the robot samples the surface of an object.

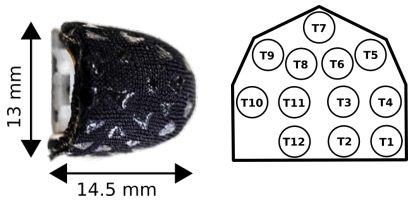
A. The Objects

The objects, as shown in Fig. 4, are wooden geometric shapes. The objects are deliberately selected to have similar shapes, this helps in better evaluating different exploration strategies. For example, the difference between the object in Fig. 4(a) and the object in Fig. 4(b) is that one has a smooth arched surface and the other has a saw-tooth shaped surface, respectively. Similarly, objects in Fig. 4(c) and Fig. 4(d) present more complex surfaces one with a smooth peak and trough and the other having a similar general shape but defined by a sharp peak and a ridge. Objects in Fig. 4(e) and Fig. 4(f) have same shape except for the bottom edges: one has a straight edge while the other has a curved one. Moreover, these two objects have large flat surfaces. This selection allows us to test the exploration strategy developed for capturing the general shape as well as detailed features of the object being explored.

B. Workspace Exploration

In order to explore the surface of an object without a priori information the robot needs to make contact with the object surface at arbitrary locations. To make the problem tractable, we define a volume of space that the robot will search for the object, which we will refer to as the workspace henceforth. In our experiments we used a $13\,\text{cm} \times 10\,\text{cm} \times 6\,\text{cm}$ workspace. The volume of the workspace is arbitrary as long as it is larger than the volume of the object. We make the assumption that the object does not move during the exploration.

The robot uses a tapping strategy to detect a contact event. The tapping strategy ensures that the robot finger is not damaged during exploration. In this section we give the details of how we detect a contact event, which is followed by a description of the tapping strategy.



(a) Finger (b) Taxel map

Fig. 5. The iCub fingertip is equipped with a tactile system made of 12 sensing units. The schematic representation on the right shows the approximate distribution of the taxels.

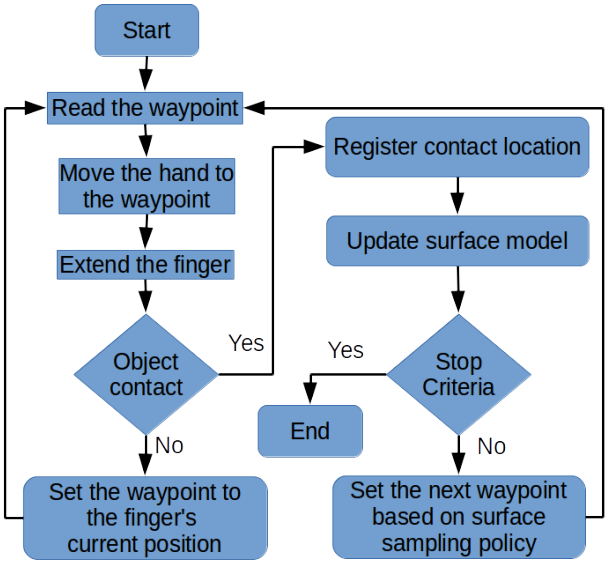


Fig. 6. A flow chart showing the object-surface sampling strategy.

1) Contact Event Detection: We use the index finger of the robot to make contact with the object. As shown in Fig. 5(a), the finger is $14.5\,mm$ long, and $13\,mm$ wide. Each finger is equipped with 12 taxel tactile sensors. The tactile sensors use the capacitive principle of transduction [22]. An applied force changes the capacitance of the taxel which is used as a proxy for the applied force. Each taxel outputs a value between 0 and 250, with zero corresponding to no contact. To detect a contact we learned a correlation between the 12 tactile sensor outputs and an applied force. A contact is defined when the force value exceeds a given threshold.

2) Surface Location Detection: At the start of the exploration an object is placed in the workspace defined earlier. Then the robot is commanded to sample a location of interest in the x-y plane. We will refer to the location of interest as a waypoint. Since we do not have a priori knowledge of the height of the object, the height of the waypoint is set to the maximum height of the workspace volume. As shown in Fig. 6, the robot moves the finger to the waypoint. Then the robot extends its finger downward to detect a contact. If no contact is detected when the finger is fully extended,

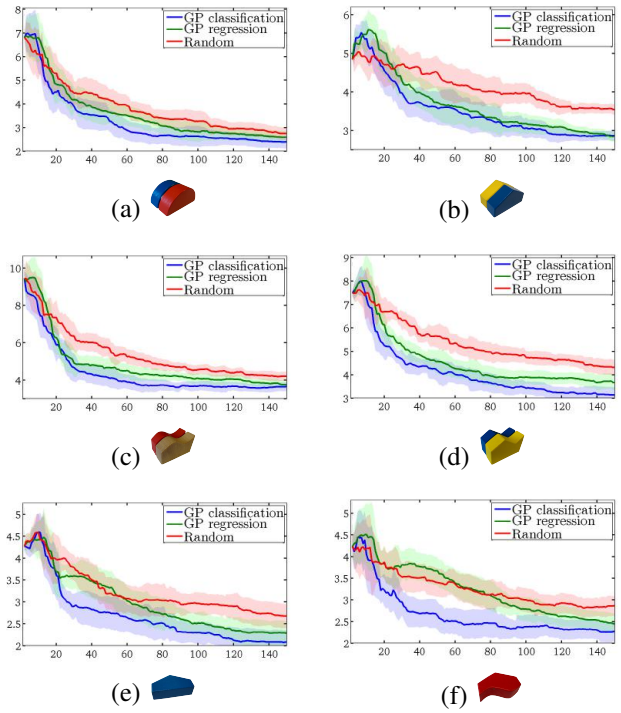


Fig. 7. Average performance of different sampling policies. The horizontal axis is the number of locations sampled. The vertical axis plots the root mean squared error between the benchmark mesh and the mesh generated as new contact locations are sampled. The shaded area is the standard deviation. It shows that the proposed method based on GP classification performs better.

the robot sets the waypoint to the current location of the finger and retracts its finger. This process is repeated until the finger makes a contact with a surface – either the object or the table. When a contact is detected, the location of the contact is registered. At this point the robot updates the surface model and queries it for the next location to be sampled. This processes is repeated until a target criterion is reached. The accompanying video shows the robot sampling a surface using this strategy. We compare three surface sampling policies: 1) random selection, 2) uncertainty in the object model constructed using GP regression, and 3) the GP classification approach proposed in this paper.

V. Results

A. Average Performance

We sampled the surface of each object in a grid of points with a $2.5\,mm \times 2.5\,mm$ cell size. This data serves as the benchmark for evaluating our algorithms. In order to be able to evaluate the results across different exploration strategies, we used the dataset to reconstruct the surface of the objects. We reconstruct the surface by fitting a mesh to the data using the MATLAB implementation of the natural neighbor algorithm[2]. Since the surface generated using the benchmark data is the best the robot can do, we use it as the ground truth.

[2]The natural neighbor algorithm[23] is based on Voronoi tessellation of a discrete set of points. The value of the function is computed by adding a new Voronoi cell in the query point. The value of the function in the query point is computed by assigning weights to the neighbor points proportional to the area that is taken by the new cell.

Figure 2 shows an object and its corresponding surface mesh. Note that we use the natural neighbor algorithm instead of Gaussian regression to produce the surface mesh because it is deterministic.

When the robot samples a new location in the workspace, it constructs a mesh of the surface using the contact points sampled thus far. The surface mesh is compared with the surface mesh generated with the benchmark data. For each object, we sampled the object surface using three strategies, namely, GP classification, GP regression, and random selection of points. Each strategy was run ten times. Figure 7 shows the root mean squared error (RMSE) between the mesh generated from the benchmark data and the mesh generated from the points collected at each iteration. The RMSE is plotted on the vertical axis, which is averaged over the ten runs. On the horizontal axis we plot the number of locations sampled. The plots show that the GP classification method outperforms both the GP regression and the random selection methods.

B. Contact Location Distribution

The RMSE gives a numerical evaluation of the performance of the proposed method. It is also possible to evaluate our method by inspecting the distribution of the contact locations sampled with the different techniques. Figure 8 shows the distribution of 150 locations sampled on each object. We can see that the algorithm proposed in this paper produces points that are more densely concentrated on and around the object. Notice that, in contrast, the strategy based on GP regression gives equal importance to all points thus converging to a uniform sampling of the workspace. Since sampling a location on the surface of an object is time consuming, by focussing on the object, the proposed method will take less time to capture the shape of the object. The attached video visualizes the surface reconstruction as the robot samples an object.

VI. Conclusions and Future Work

We presented an active exploration strategy to reconstruct the shape of an object using tactile feedback. The tactile sensors are used to detect contact with the object, thereby, generating a point cloud of contact locations, which is used to construct the shape of the object. The main contribution of the paper is the way we use a probabilistic classification – GP classification – to guide the exploration so that more points are sampled on the object and around its perimeter. In the proposed method, the robot, iteratively, makes contact with an object. At each iteration the data collected is used to construct a probabilistic model of the object's surface. The robot uses these probabilities to guide the surface exploration. We demonstrated with experiments conducted on a real robot equipped with tactile sensors that our approach performs better than random selection and previous work based on GP regression. We demonstrated that with our approach the sampled locations are densely concentrated on and around the object, while in the GP regression approach it performs a uniform sampling of the workspace.

In this work the tactile sensors on the robot have only been used to detect contact. However, the tactile data can provide useful information on object features such as edges and local surface curvature that could be used to guide the exploration. In this work, we deliberately avoided vision. However, the stereo system on the robot can provide cues on the object location that can be used to bias the exploration and further reduce the time required to localize and sample the object. In the future we will also relax the object immobilization assumption.

REFERENCES

[1] A. Bicchi and V. Kumar, "Robotic Grasping and Contact: A Review," *Proc. IEEE Int. Conf. Robot. Autom.*, 2000.

[2] S. e. a. Izadi, "Kinect Fusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera," in *Proc. ACM Symp. User interface Softw. Technol.*, 2011.

[3] J. Rock, T. Gupta, J. Thorsen, J. Gwak, D. Shin, and D. Hoiem, "Completing 3D Object Shape from One Depth Image," in *Comput. Vis. Pattern Recognit.*, 2015.

[4] P. Allen, A. Miller, P. Oh, and B. Leibowitz, "Using Tactile and Visual Sensing with a Robotic Hand," *Proc. Int. Conf. Robot. Autom.*, 1997.

[5] Z. Yi, R. Calandra, F. Vegia, H. van Hoof, T. Hermans, Y. Zhang, and J. Peters, "Active Tactile Object Exploration with Gaussian Processes," in *IEEE Int. Conf. Intell. Robot. Syst.*, 2016.

[6] J. A. Fishel and G. E. Loeb, "Bayesian exploration for Intelligent Identification of Textures," *Front. Neurorobot.*, 2012.

[7] R. Martins, J. F. Ferreira, and J. Dias, "Touch Attention Bayesian Models for Robotic Active Haptic Exploration of Heterogeneous Surfaces," in *IEEE Int. Conf. Intell. Robot. Syst.*, 2014.

[8] N. Jamali and C. Sammut, "Majority Voting: Material Classification by Tactile Sensing Using Surface Texture," in *IEEE Trans. Robot.*, 2011.

[9] C. M. Oddo, M. Controzzi, L. Beccai, C. Cipriani, and M. C. Carrozza, "Roughness Encoding for Discrimination of Surfaces in Artificial Active-Touch," *Robot. IEEE Trans.*, 2011.

[10] A. M. Okamura, "Feature Detection for Haptic Exploration with Robotic Fingers," *Int. J. Rob. Res.*, 2001.

[11] U. Martinez-Hernandez, G. Metta, T. J. Dodd, T. J. Prescott, L. Natale, and N. F. Lepora, "Active Contour Following to Explore Object Shape with Robot Touch," in *World Haptics Conf.*, 2013.

[12] H. Liu, K. C. Nguyen, V. Perdereau, J. Bimbo, J. Back, M. Godden, L. D. Seneviratne, and K. Althoefer, "Finger Contact Sensing and the Application in Dexterous Hand Manipulation," *Auton. Robots*, 2015.

[13] L. Natale and E. Torres-Jara, "A Sensitive Approach to Grasping," *Proc. Sixth Int. Work. epigenetic Robot.*, 2006.

[14] K. C. Nguyen and V. Perdereau, "Fingertip Force Control Based on Max Torque Adjustment for Dexterous Manipulation of an Anthropomorphic Hand," in *IEEE Int. Conf. Intell. Robot. Syst.*, 2013.

[15] D. Montana, "The Kinematics of Contact and Grasp," *Int. J. Rob. Res.*, 1988.

[16] S. Dragiev, M. Toussaint, and M. Gienger, "Gaussian Process Implicit Surfaces for Shape Estimation and Grasping," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011.

[17] M. Bjorkman, Y. Bekiroglu, V. Hogman, and D. Kragic, "Enhancing Visual Perception of Shape through Tactile Glances," in *IEEE Int. Conf. Intell. Robot. Syst.*, 2013.

[18] C. E. Rasmussen, *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[19] I. Steinwart and A. Christmann, *Support Vector Machines*. Springer Science & Business Media, 2008.

[20] A. Tacchetti, P. K. Mallapragada, L. Rosasco, and M. Santoro, "GURLS: A Least Squares Library for Supervised Learning," *J. Mach. Learn. Res.*, vol. 1, 2013.

[21] G. Metta, G. Sandini, D. Vernon, L. Natale, and F. Nori, "The iCub Humanoid Robot: An Open Platform for Research in Embodied Cognition," *Proc. 8th Work. Perform. Metrics Intell. Syst.*, 2008.

[22] A. Schmitz, P. Maiolino, M. Maggiali, L. Natale, G. Cannata, and G. Metta, "Methods and Technologies for the Implementation of Large-Scale Robot Tactile Sensors," *IEEE Trans. Robot.*, jun 2011.

[23] R. Sibson *et al.*, "A brief description of natural neighbour interpolation," *Interpreting Multivariate Data*, 1981.

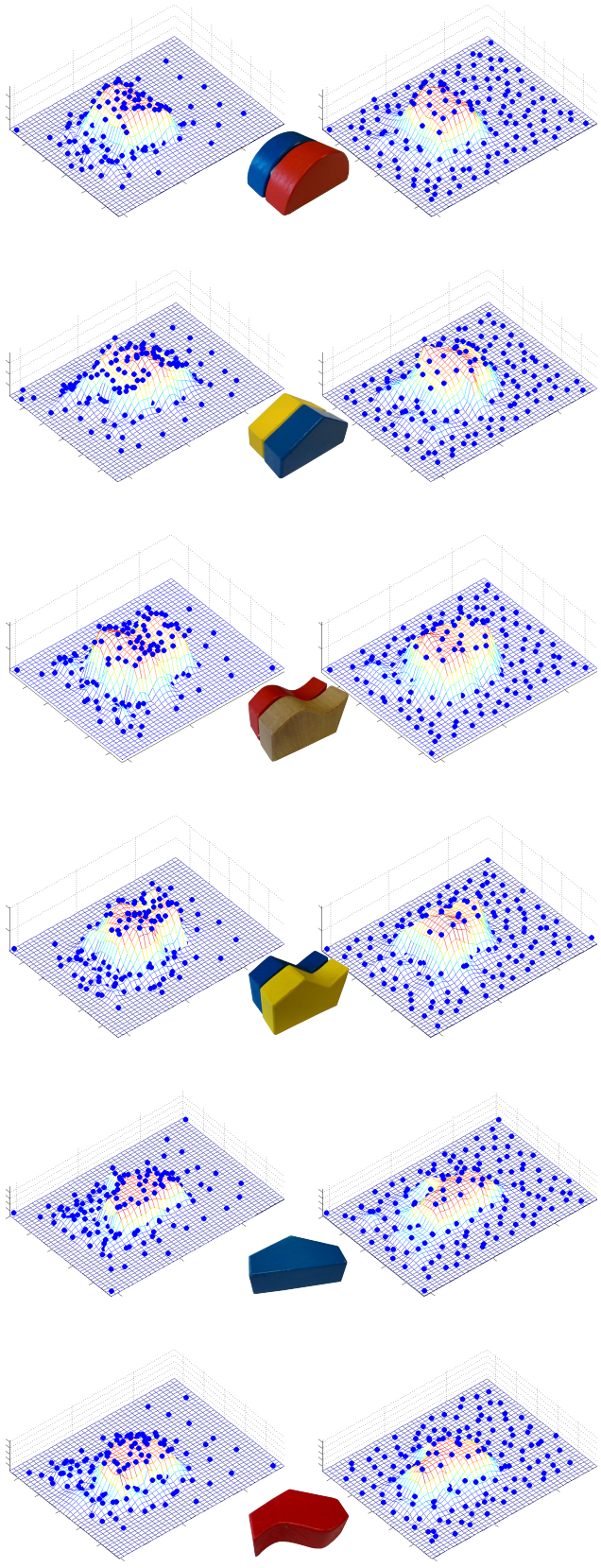


Fig. 8. Distribution of the sampled points. The left column shows the results for the approach proposed in this paper. The middle column shows the object. The right column shows the results for the exploration based on GP regression.