GPAtlasRRT: A Local Tactile Exploration Planner for Recovering the Shape of Novel Objects

Carlos Rosales* and Federico Spinelli†

Centro di Ricerca E. Piaggio,
Università di Pisa, Pisa, Italy
**carlos.rosales@for.unipi.it*
†*fspinelli@gmail.com*

Marco Gabiccini

Dipartimento di Ingegneria Civile e Industriale,
Largo Lucio Lazzarino 1, Universita di Pisa, 56122 Pisa PI, Italy
marco.gabiccini@unipi.it

Claudio Zito‡ and Jeremy L. Wyatt§

IRLab, CN CR, School of Computer Science,
University of Birmingham,
Birmingham B15 2TT, UK
‡*C.Zito@cs.bham.ac.uk*
§*J.L.Wyatt@cs.bham.ac.uk*

Received 1 April 2017
Accepted 15 January 2018
Published 27 February 2018

Touch is an important modality to recover object shape. We present a method for a robot to complete a partial shape model by *local tactile exploration*. In local tactile exploration, the finger is constrained to follow the local surface. This is useful for recovering information about a contiguous portion of the object and is frequently employed by humans. There are three contributions. First, we show how to segment an initial point cloud of a grasped, unknown object into hand and object. Second, we present a local tactile exploration planner. This combines a Gaussian Process (GP) model of the object surface with an AtlasRRT planner. The GP predicts the unexplored surface and the uncertainty of that prediction. The AtlasRRT creates a *tactile exploration path* across this predicted surface, driving it towards the region of greatest uncertainty. Finally, we experimentally compare the planner with alternatives in simulation, and demonstrate the complete approach on a real robot. We show that our planner successfully traverses the object, and that the full object shape can be recovered with a good degree of accuracy.

Keywords: Shape modeling; tactile exploration.

§Corresponding author.

1. Introduction

Recovery of the properties of a new object is a basic task in robot manipulation. Properties of interest include object shape, texture, friction coefficients, elasticity, plasticity, etc. Because the object is initially unknown, the sensing strategy must be active, i.e., the robot must adapt its sensing actions based on the results so far. A popular modality for active sensing is vision. Active vision has been studied in depth for the problem of shape and pose recovery for robot grasping.[1–5] Humans, however, also use active tactile sensing.[6] There is a variety of work on tactile exploration for robot manipulation.[7–13] The goal of this paper is to extend the set of available techniques for guiding tactile exploration to recover surface shape.

The requirements for active tactile perception were authoritatively spelled out by Bajcsy and co workers[14] in the 1980s. Early tactile perception algorithms date back to the same period.[15–18] Active touch, however, lags behind active vision for two reasons. The first reason concerns robot hardware. Standardized touch sensors are still not widely available and typically have to be handcrafted or modified for the specific robot and task. The second reason is intrinsic to touch itself, which requires the mechanical interaction of the sensor and the object being perceived. This inevitably leads to unexpected perturbations of the sensor and object, which in turn require rather complex control of the ongoing movement of the sensor: a requirement which is absent in vision.

In this paper, we suppose that vision has already provided some partial information about the shape of the object. Given this initial, incomplete, surface model, our method plans a sequence of touches. The planning relies on the ability of the robot to form hypotheses about the shape of the hidden parts of the object. The robot then attempts to touch the surface, so as to refine these hypotheses. Whether or not a contact is made, the information gained improves the model of the object shape.

Tactile information is sparse, and so refining beliefs about the shape must use data efficient inference. We follow others working on tactile estimation of surface shape[9,19,20] by employing Gaussian Process (GP) inference.[21] This produces data efficient, nonlinear regression estimates of the object's surface shape. It also predicts the variance in these surface estimates. The surface is implicitly defined, being the 0 level set of an *unknown* function. This implicit surface representation is also well established.

Given this combination of a representation and an inference method, the remaining issue is how to explore the object surface so as to generate the data points. The criterion that we use to drive exploration is reduction of uncertainty in object shape, again similar to that deployed in the previous work.

Where we make our contribution is in the fact that where others define the exploratory actions as poking or grasping actions defined in an essentially two dimensional (2D) workspace (pushing vertically down onto the object surface from above; grasping while moving along the object's vertical axis; moving a finger in the 2D plane to a touch point, or following an edge), we address the problem of following

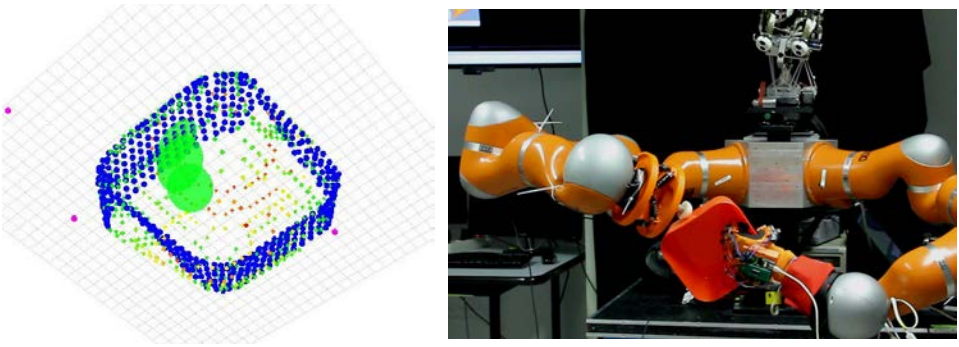


Fig. 1. (Left) The GPAtlasRRT strategy suggests touches (light green colored disks) on the predicted surface. The blue points show the initial partial reconstruction from a depth camera that the robot uses to guide tactile search. The predicted surface is also shown as points colored from green to red. Green indicates high uncertainty in the surface prediction, and red indicates low uncertainty. (Right) Our Vito robot executing a step of tactile exploration.

the object surface as it curves through three dimensional (3D) space. Thus, our experimental scenario is to have a grasped object that must be explored on as many sides as is kinematically feasible (Fig. 1). This first means planning poking actions that have a full 6 degrees of freedom (DoF) (position and poke orientation relative to the target touch point and surface normal). In addition, in our problem, the robot can "trace" the surface, either by sliding, or by making a closely spaced series of touches that follow the surface as it curves. This requires that we interpret the implicit surface as a manifold, which constrains the configuration space. We then build a tactile exploration planner using recent methods for sample based exploration on manifolds.[22] The benefit of our approach is the first tactile exploration algorithm that can plan to cover the surface of a 3D object.[a] We also demonstrate a real robot system that can perform this exploration while the object is being grasped by a soft hand.

In more detail, to achieve this, given an implicitly defined predicted surface, we build a collection of charts (an atlas) that model the shape, and use these to perform a search across the object, driven towards areas of high uncertainty in the implicit surface. The construction of the atlas follows, as well as drives, the exploration. Specifically, we search for points on the estimated surface that have a variance larger than some pre specified threshold. The expansion and planning process is repeated after execution of each touch. By repeated touches, the robot will converge on an estimated surface, such that any point on it has low variance. The terminating condition is met when no candidate for the next best touch is found by the GP AtlasRRT algorithm, which means that the object shape prediction meets the requirement on variability. Naturally, the smaller the threshold (i.e., the more accurate

[a] Assuming that the object cannot be grasped two sides at a time on all parts of the object, as utilized by Bjorkman *et al.*[9]

the model is required to be), the higher the number of touches required to converge. This threshold is the only input to the devised strategy, given either by a higher level module or the user.

The structure of the paper is as follows. In Sec. 2, we first review the previous work related to tactile exploration and object shape representation. In Sec. 3, we clearly state the problem we aim to solve and in Sec. 4, we present the envisioned approach for its solution. The experimental results and their discussion are presented in Sec. 5. Finally, conclusions and points deserving further attention are given in Sec. 6.

2. Related Work

One of the first attempts to exploit active tactile exploration with passive stereo vision for object recognition was proposed by Allen.[23] In that paper, a rigid tactile sensor was traced along the object's surface in pre defined movements. The work was later extended to develop different exploratory procedures to acquire and interpret 3D information on the surface shape.[24] The exploratory procedures were, however, selected by a human. Single finger tactile exploration strategies for recognizing polyhedral objects have also been evaluated in simulation.[25,26]

Multiple fingers have also been used for tactile sensing. Moll and Erdmann[27] presented a method for reconstructing the shape and motion of an unknown convex object using three sensorized fingers. In that approach, the object's friction proper ties must be known *a priori* and its surface must be smooth, lacking sharp edges and corners.

Tactile sensing has been used for localization rather than surface recovery. Pet rovskaya and Khatib[11] used tactile exploration to localize an object of known shape. Since full Bayesian estimation of the pose of a free body is computationally expen sive, they approximated the posterior with particles. For a well constrained object dataset, the approach performs pose estimation in under 1 s with high reliability.

Bayesian methods have also been employed in shape estimation. Meier *et al.*[28] performed tactile shape reconstruction using a Kalman filter. Efficient Bayesian inference using GPs has been used by various authors. For example, Sommer *et al.*[20] proposed a method for bimanual compliant tactile exploration that used the GP representation to smooth the noisy point data, although they did not exploit the GP representation to derive the exploratory strategy.

Dragiev *et al.*[19] presented one of the first works to employ the GP Implicit Surface (GPIS) representation for concurrent representation of the object shape and guidance of grasping actions. However, that paper utilized only the maximum *a posteriori* (MAP) estimate of the shape and thus did not utilize the ability of the GP to capture the uncertainty in the surface estimate. Later work, by the same authors,[29] offered a way to prefer regions of the model with a particular certainty level in their shape estimate and introduced the notion of explore grasp and exploit grasp primitives. In other work,[30] GPIS has been compared to a different implicit surface model, showing some of the disadvantages of GPIS when modeling features such as edges and corners.

Algorithms for selecting the sequence of touches have been developed by various authors. Bierbaum *et al.*[31] guided the tactile exploration by using Dynamic Potential Fields for motion guidance of the fingers. They showed that grasp affordances can be generated from geometric features extracted from the contact points.

Bjorkman *et al.*[9] showed how to build object models with a small number of tactile actions (each action involving multiple simultaneous touches by several tactile arrays) with the aim of categorization, rather than shape recovery. They employed the GPIS representation mentioned above, with the kernel used by the GP being the thin plate covariance function derived by Williams.[32] A set of predefined tactile glances are performed on the object: however, these are not updated as the object model is refined.

Bjorkman's approach is what we term *global tactile exploration* rather than *local tactile exploration*. By local, we mean that the touch choices are constrained so that subsequent touches are close to one another and to the already explored surface. The area considered for exploration grows outwards until a suitably uncertain point is found. This local exploration is very different from a global exploration strategy, in which any touch action can be considered. There is no inherent benefit to either approach, but local exploration allows us to define a series of touches across a contiguous area of hypothesized surface. Local exploration is a strategy often employed by humans. Both local and global strategies are important, and local tactile exploration has also been considered by others,[33,34] but this is the first paper in which a local exploration strategy is formulated for full exploration and recovery of a complete surface as it curves through 3D space.

There are also other smaller differences with Bjorkman's work: the space in which the next best exploratory action is computed, the grain size to compute the predicted shape, and the terminating condition for the overall algorithm. Bjorkman *et al.*[9] for example, drew the exploratory actions from a discretization of the vertical axis of the workspace and the approach angle. This works because the objects are placed upright on a table. But, it means that the actions are extrinsic to the shape model. This is not suitable for exploration while the object is being held by the robot. Neither is there any guarantee that the contact will be on a particular location on the object surface. Moreover, since they are interested in a model that is useful for categorization, they proposed pointwise curvatures to make it affine invariant, for which a fine grained explicit representation is required. We also compute the predicted shape with a coarse grain for collision avoidance purposes, an issue that is not considered in that work. Finally, the number of touches in Bjorkman's work is subject to an absolute limit. The set is ordered according to the closest point on the implicit function with high variance. In contrast, we set a maximum acceptable uncertainty for the predicted shape. As a consequence, we can continue to explore until the shape is sufficiently well known.

More recently, active touch using a GP model has been developed by Jamali *et al.*[35] That paper uses a combination of GP regression and GP classification to pick the next best sample point for a finger. This is driven to where the model has the

lowest confidence in its prediction. One difference with our approach is that the sampling points are specified in the x, y plane and not in full 3D space. This some what simplifies their path planning problem, as they do not have to calculate a path over the surface. They also differ in that they do not employ an implicit surface model. In addition, their system is unimanual.

A similar approach is that of Yi *et al.*, who used a single finger probe to explore objects that are fixed to a surface. The next best point is the one that has the greatest variance in the height of the predicted object surface. This is similar to Jamali *et al.* if the exploration was dominated by the GP regression model. Our method differs from this work in a similar way to that in which it differs from Jamali *et al.*[35] Our problem is to plan an exploration path on a 3D surface involving a sequence of contacts at a time, not to select a single next best touch point normal to a plane.

Finally, closer again to our approach is that of Matsubaru *et al.*[33] In this, a GPIS model is employed, together with a planner that accounts for the trade off between the travel distance between touches, and the uncertainty in the surface at the pro posed touch location. Thus, it is the first example of an active touch planner that is local while still being driven towards areas of uncertainty. The main restriction of that work is its restriction to a 2D model of the 3D object shape (its projection in the vertical plane). The former allows the use of a grid based discretization of the workspace as the space within which touches are chosen. A similar approach to the same problem has been taken by Tosi *et al.*,[36] who pose the planning as a joint optimization problem, again restricting shape recovery to the 2D outline of an object.

In our work, we instead focus on the problem of how to plan a path of touches for the robot across a 3D surface. To solve this, we exploit rapidly exploring random trees (RRT) based planning in continuous but constrained configuration spaces, and approach which is arguably more scalable, although we do not make a comparison here. Other work that employs local tactile exploration is,[34,37] which uses a discrete Bayesian model of the properties of edges of objects together with active exploration to follow those edges, the most complex of which is arguably a volute ridge. It does not, however, model a complete object surface.

3. Problem Statement

Knowledge of object shape is necessary for many manipulation tasks.[18] The best shape representation depends on the precise task, but there are many generally desirable properties. These include, among others: accuracy, compactness, an intu itive parametrization, local support, affine invariance, an ability to handle arbitrary topologies, guaranteed continuity, efficient rendering and support for efficient colli sion detection. Since we are concerned with shape recovery for arbitrary novel objects, the capacity to represent an arbitrary topology while retaining guaranteed continuity is desirable. Implicitly defined surfaces have these properties.

There are several ways to define an implicitly defined surface, e.g., via algebraic equations, blobby models, or variational surfaces. These classical representations do

not include any measure of shape uncertainty. There is where the work by Williams and Fitzgibbon[32] comes in, introducing the notion of a GPIS. This is not the only representation to account for uncertainty, but it meets the requirements for a good shape representation in robotics, as discussed above. Section 3.1 introduces the notation for GPIS.

The surface estimate, when using GPIS, is simply the mean value of a GP, which is the 0 level set of an implicitly defined manifold. Henderson *et al.*[38] provide a way to recover the implicitly defined surface via continuation techniques. That work has been extended in a number of different directions, including one of the particular interests for local exploration. The AtlasRRT algorithm is a path planning method for constrained manipulators.[22] It combines continuation techniques for surface recovery with RRT[39] for path generation. This combination allows the computation of paths in a constrained configuration space, i.e., a manifold (such as a surface) embedded in an ambient space (such as the workspace). However, the AtlasRRT, as employed to date, makes no use of uncertainty in the manifold. By combining the AtlasRRT algorithm with the concept of uncertainty as modeled with GPIS, we can derive a powerful planner for traversing an uncertain surface. This is the central technical contribution of this paper. Section 3.2 describes the basic idea behind the AtlasRRT algorithm.

The final ingredient required for bimanual object exploration is the equipment necessary to simultaneously grasp an object with one hand whilst exploring it with the other. In Sec. 3.3, we enumerate the considerations for the hardware that is to execute tactile exploration, as well as possible limitations. Finally, with all these ingredients in mind, we formally define our problem in Sec. 4.

3.1. *Gaussian process implicit surfaces*

A surface embedded in a 3D Cartesian space can be regarded as the 0 level set of a family of surfaces defined by an implicit function $F(\mathbf{x}, y) = 0$, where $F : \mathbb{R}^{3+1} \rightarrow \mathbb{R}$, with coordinates $\mathbf{x} \in \mathbb{R}^3$ and parameter $y \in \mathbb{R}$. Under the assumption that the implicit function theorem holds, it can be expressed, at least locally, as $y = f(\mathbf{x})$, with $F(\mathbf{x}, f(\mathbf{x})) = 0$. The surface of interest arises when we set $y = 0$, i.e., when we define the 0 level set. The value of f (i.e., y) is positive and increasing as we move outwards from the surface (∇F), and negative and decreasing as we move further inside the object ($-\nabla F$).

A GP "is a collection of random variables, any finite number of which have a joint Gaussian distribution".[21] It is completely specified by a mean, $m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$, and a covariance, $k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))]$, function, where $\mathbb{E}(\cdot)$ is the expected value of a real process, such that we can write

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')). \tag{1}$$

Now, let $\mathcal{S}$ be a set of tuples $s_i = (\mathbf{x}_i, \sigma_i, y_i)$ with $i = 1, \ldots, n$. Here, the $\mathbf{x}_i$ are points in the Cartesian workspace, σ_i are corresponding noise parameters of the tactile observations,[b] and y_i is the target value for the implicit function (either -1, 0 or $+1$). The set S constitutes the tactile observations that are used as the training set for the GP. We specify $m(\mathbf{x}) = 0$ to yield the model

$$\mathbf{y} \sim \mathcal{N}(\mathbf{0}, K(\mathcal{X}, \mathcal{X}) + \boldsymbol{\sigma}^\top I \boldsymbol{\sigma}), \tag{2}$$

where $\mathcal{N}(\cdot, \cdot)$ denotes a normal distribution parametrized by mean and variance; $\mathcal{X}$ corresponds to the inputs from the training set $\mathcal{S}$; $K(\cdot, \cdot)$ is the covariance matrix formed from elements $k_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, for all pairs of input points $i, j : \mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$; $\boldsymbol{\sigma}$ is the n dimensional vector corresponding to the noise of the ith observation;[c] and finally, $\mathbf{y}$ is the n dimensional vector of target outputs.

The purpose is to predict a vector of target values, $\mathbf{y}_*$, given test inputs $\mathcal{X}_*$. To achieve this, Eq. (2) can be block expanded[d,21] to give

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{y}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K + \boldsymbol{\sigma}^T I \boldsymbol{\sigma} & K_* \\ K_*^\top & K_{**} \end{bmatrix}. \right), \tag{3}$$

The two predictive equations can then be derived via algebraic manipulation:

$$\mathbf{y}_* = K_*^\top [K + \boldsymbol{\sigma}^T I \boldsymbol{\sigma}] \ ^1\mathbf{y}, \tag{4}$$

$$\mathbb{V}(\mathbf{y}_*) = K_{**} - K_*^\top [K + \boldsymbol{\sigma}^T I \boldsymbol{\sigma}] \ ^1K_*. \tag{5}$$

The first is the vector of predicted values of the implicit function, and the second is the vector of the variances in those predictions. If we wish to make a prediction for a single test input $\mathbf{x}_*$, we follow Ref. 21 in further simplifying the notation for the covariances. In that case, there is a vector of covariances between the test point and each of the training inputs, denoted $\mathbf{k}(\mathcal{X}, \mathbf{x}_*)$. We will use this later on.

The key choice in using a GP is the choice of kernel for specifying the covariance between two points in the input space. Intuitively if input points $\mathbf{x}_i, \mathbf{x}_j$ are close together then, in order to have a smooth function, they should strongly covary. Conversely, as the distance between input points $\mathbf{x}_i, \mathbf{x}_j$ increases, their covariance tends to zero. We utilize the idea, proposed by Williams,[32] to use the thin plate kernel

$$k(r) = 2r^3 - 3Rr^2 + R^3, \tag{6}$$

with $r = \|\mathbf{x} - \mathbf{x}'\|_2$ and R being the largest r in the training set. This training set only consists of points on the object surface. To aid training of the implicit function, we therefore extend the training set to be the composition three sets. First, there are points on the surface $\mathcal{S}^0$, with tuples of the form $s_i = (\mathbf{x}_i, \sigma_i, 0)$. Then, there are points outside the surface, $\mathcal{S}^+$, with tuples of the form $s_i = (\mathbf{x}_i, 0, +1)$ and finally there are points

[b] We set this to 10 mm for visual and 5 mm for tactile sensing after experimentation.
[c] Therefore, I is an $n \times n$ identity matrix.
[d] For simplicity, we drop the arguments of the matrices such that $K \quad K(\mathcal{X}, \mathcal{X})$, $K_* \quad K(\mathcal{X}, \mathcal{X}_*)$ and $K_{**} \quad K(\mathcal{X}_*, \mathcal{X}_*)$.

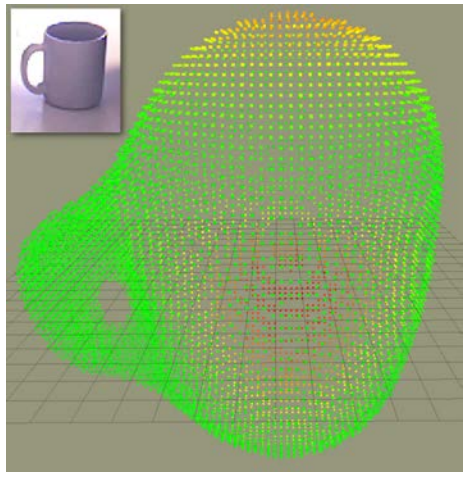


Fig. 2. Gaussian Implicit Surface, obtained from a mug (top-left corner) and sampled with a box-grid evaluation with fairly high point density. Each point has a predicted target $y_* \simeq 0$ and is colored accordingly to its associated predicted variance, from red (high variance) to green (low variance).

inside the surface, $\mathcal{S}$, with tuples of the form $s_i = (\mathbf{x}_i, 0, -1)$. Thus, the training set $\mathcal{S} = \mathcal{S}^0 \cup \mathcal{S}^+ \cup \mathcal{S}$.[e]

We are now able to predict the target $y_* = f(\mathbf{x}_*)$ and its variance $\mathbb{V}[f(\mathbf{x}_*)]$ for any given test point $\mathbf{x}_*$ in the workspace, given the training set $\mathcal{S}$. To find the implicit surface, we need to exhaustively evaluate y_* for each point $\mathbf{x}_*$ in a 3D box grid containing the object. The predicted surface points $\mathbf{x}_*$ are those where $y_* \simeq 0$. Figure 2 shows a GPIS, for the pictured mug, sampled with a box grid evaluation. We can use this to find candidate surface points for tactile exploration.

We must also find the best direction for the finger to approach the surface. A good choice is the predicted surface normal at the candidate point. In our case, the normal is parallel to the gradient of the function. If we consider the posterior mean of the GP given in Eq. (4) for a single test point, we have

$$\begin{aligned} f(\mathbf{x}_*) &= \mathbf{k}(\mathcal{X}, \mathbf{x}_*)^\top [K + \boldsymbol{\sigma}^T I \boldsymbol{\sigma})] \ {}^1\mathbf{y} \\ &= \mathbf{k}(\mathcal{X}, \mathbf{x}_*)^\top \boldsymbol{\alpha}. \end{aligned} \tag{7}$$

Note that the vector $\boldsymbol{\alpha}$ is constant for a given training set $\mathcal{S}$, whereas the vector $\mathbf{k}(\mathcal{X}, \mathbf{x}_*)$ gathers the covariance values between the test point and the training set being the only term depending on the test point. Therefore, the gradient evaluated at $\mathbf{x}_*$ is

$$\frac{\partial f(\mathbf{x}_*)}{\partial \mathbf{x}_*} = \frac{\partial \mathbf{k}(\mathbf{x}_*, \mathcal{X})}{\partial \mathbf{x}_*} \boldsymbol{\alpha}, \tag{8}$$

[e]Without loss of generality, but with a slight gain in efficiency and parameter tuning, we can also work in a normalized and offset-free space, using as scale the larger distance and the centroid from the training set. This way, for instance, R becomes a fixed parameter, as well as the S^+ and S^- sets, a trick also exploited in Ref. 40. Of course, the model exploitation requires a re-scaling and re-centering processing step.

which boils down to evaluating, for each combination of test and training point, the derivative of the thin plate covariance function

$$\frac{\partial k(r)}{\partial r}\frac{\partial r}{\partial \mathbf{x}_*} = [6r(r-R)]\frac{\mathbf{x}_i - \mathbf{x}_*}{\|\mathbf{x}_i - \mathbf{x}_*\|_2},$$
$$\frac{\partial k_*}{\partial \mathbf{x}_*} = 6(r-R)(\mathbf{x}_i - \mathbf{x}_*), \tag{9}$$

for all $i : \mathbf{x}_i \in \mathcal{X}$. Consequently, the normal at the test point, $\mathbf{n}_*$, is obtained dividing the gradient by its magnitude. Equation (8) is equivalent to $\nabla f(\mathbf{x})$.

3.2. *Defining an Atlas of an implicitly-defined surface*

How might we represent the implicit surface in such a way that we can easily create paths across it using standard path planning algorithms? The insight comes from the fact that the surface is simply one example of a more general phenomenon: a smooth manifold embedded in some higher dimensional space. Henderson[38] gave a precise method to model such manifolds via a collection of *disks*. Each disk lies on the tangent plane to the manifold at some point on the manifold. In our case, the manifold is the surface of the object, and it is embedded in the Cartesian workspace. We also refer to the disks as charts, which we can later use for path planning, and thus we also refer to a collection of disks covering the surface as an atlas. The creation of the atlas starts with selection of an initial point, $\mathbf{x} \in \mathbb{R}^3$, known to be on the surface, $f(\mathbf{x}) = 0$ (or very close and projected onto it), which gives the center of the first disk. Adjoining disks are created from this first one. The method continues iteratively until all disks are surrounded by neighbors, and the atlas thus provides a complete coverage of the shape. This concept has been widely used, including for obtaining representations of constrained configuration spaces,[41] such as object surfaces. We now give some details.

First, recall that our implicit surface is defined by the equality constraint f that holds for all points in the set $\mathcal{X}$, of points on the object surface

$$\mathcal{X}, = \{\mathbf{x} \in \mathbb{R}^3 : f(\mathbf{x}) = 0\}. \tag{10}$$

For any point $\mathbf{x}_i \in \mathcal{X}$, we can find its tangent space, i.e., the tangent plane to the surface at $\mathbf{x}_i$. The matrix $\mathbf{\Phi}_i$ is the basis of the tangent space for $\mathbf{x}_i$. This therefore defines the mapping of points from this tangent space into $\mathbb{R}^3$. Matrix $\mathbf{\Phi}_i$ satisfies

$$\begin{bmatrix} \nabla f(\mathbf{x}) \\ \mathbf{\Phi}_i^\top \end{bmatrix} \mathbf{\Phi}_i = \begin{bmatrix} 0 \\ I \end{bmatrix}, \tag{11}$$

where $n = 3$ and $k = 2$, hence I is the 2×2 identity matrix, and $\mathbf{\Phi}_i$ is a 3×2 matrix. Now, let $\mathbf{u}$ be the coordinate of a point in the tangent space of $\mathbf{x}_i$. It can be mapped to a point $\mathbf{x}'_i \in \mathbb{R}^3$ as follows:

$$\mathbf{x}'_i = \mathbf{x}_i + \mathbf{\Phi}_i \mathbf{u}_i. \tag{12}$$

Next, we find the orthogonal projection of $\mathbf{x}'_i$ onto the object surface, giving $\mathbf{x}_j$. This is achieved by solving the system

$$\begin{cases} f(\mathbf{x}_j) = 0, \\ \mathbf{\Phi}^\top(\mathbf{x}_j - \mathbf{x}'_i) = 0, \end{cases} \tag{13}$$

where the first equation compels the point x_j to lie on the object surface, and the second equation compels the projection from the tangent plane to the surface to be orthogonal. We denote the solution to this system with the function $\mathbf{x}_j = \psi_i(\mathbf{u}_i)$, and we use a gradient descent like method to solve it. The new point $\mathbf{x}_j$ is the surface point defining the center of the next disk.[f]

Thus, given a point $\mathbf{x}_i \in \mathcal{X}$, one can build a chart $\mathcal{C}_i$ that allows us to obtain a new point $\mathbf{x}_j \in \mathcal{X}$. Then, this new point can be used to generate a new chart $\mathcal{C}_j$, and so on. In order to avoid the parametrization of areas already covered by other charts, they are intersected according to their validity region, introducing the notion of bounded and unbounded charts, depending on whether they have been intersected from all direc tions or not. For instance, the initial chart is by definition unbounded. This coordi nation process yields the concept of an atlas $\mathcal{A}$: a collection of properly coordinated charts, that completely covers the manifold when there are no unbounded charts.

The manifold $\mathcal{X}$ is smooth everywhere, and without singularities. The target function $f(x)$ also exists for any point in the ambient space and for any point in the tangent spaces defined by the Atlas.

Given this machinery, we are now able to compute an atlas $\mathcal{A}$ of an implicitly defined surface $\mathcal{X}$, given a single starting point $\mathbf{x}_i$ that lies on the surface or is sufficiently close to it. How do we determine the direction in which to expand the initial chart $\mathcal{C}_i$? If we are computing an exhaustive atlas, we may choose randomly. However, if one wishes to traverse the surface of the object from one point $\mathbf{x}_i \in \mathcal{X}$ to some other point $\mathbf{x}_j \in \mathcal{X}$ while always remaining in contact with the surface, then there is no need to compute the full atlas, but only the parts covering a path that connects them.

One way to find only the necessary charts is to adapt the work of Jaillet *et al.*,[22] who successfully applied the RRT path planning technique to computing collision free paths on manifolds. We extend this so as to use the RRT to drive exploration toward uncertain regions. In other words, our atlas naturally grows towards regions of the predicted surface that need to be improved via tactile exploration.

3.3. *Equipment specification and limitations*

We now quickly specify additional constraints on the solution entailed by a practical hardware setup. We would like to recover the contact point on both fingertip and

[f]The region of a chart is defined by the choice of $\mathbf{u}$ which is typically bounded using rules about the local curvature of the manifold and the distance from the tangent space to the manifold. However, since these features are not precisely known in our scenario, we instead employ a slightly different criterion to bound each chart. We describe this later on.

object and the contact normal. There are two main suitable sensor types: (1) tactile arrays and (2) intrinsic tactile sensors.[g] The first type is composed of a grid of pressure cells of fixed area, so the point resolution is limited to the quantization of the array. This kind of sensor has been widely used due to its multi contact capability. The second type is a six axis force–torque sensor mounted in a fingertip, the shape of which permits computation of the contact point and force in closed form.[42] This is a single contact sensor with the pose resolution being typically finer than that of a tactile sensor array.

A consideration for both types is that, for tactile exploration, they need to be mounted on a robot with at least 6 DoF, to allow the exploration to happen in different orientations with respect to the explored object surface. The mobility can be increased if the object is grasped by a second robot manipulator. The object is both unknown in shape and yet requires a firm grip to resist the forces made by the tactile finger. This in turn requires that the gripper be adaptive to unexpected contacts, yet firm when the grasp is achieved.

For either sensor type, the reachable space is limited by the size of the probe. With the intrinsic tactile sensor, one can build a very small tip so as to reach small concave spaces on the object.

4. GPAtlasRRT

In the preceding section, we described how an implicit surface representation can be used to create an atlas of local charts. We also mentioned the existence of an algo rithm for creating an atlas by using an RRT path planning algorithm. We now continue with a description of how to bring these two elements together into what we term the GPAtlasRRT strategy.

The method starts from an incomplete observation of the object surface with a depth camera. To this end, we assume there is a way to segment the object from the background scene.[h] The initial observations $\mathcal{S}^0$ are used to infer the implicit surface model, $\mathcal{GP}$ and to start to build the atlas. Algorithm 1 describes how the atlas is built so as to generate candidate points for tactile exploration which will improve the implicit surface model, $\mathcal{GP}$, so as to a predefined maximum variance, $\mathbb{V}_{\max}$.

The first step is SELECTSURFACEPOINT (line 1) which randomly obtains a point $\mathbf{x}_i \in \mathcal{X}^0$ on which the first chart will be centered, invoking the CREATECHART function (line 2). The generated data structure for a chart contains: its center, $\mathbf{x}_i$; the or thonormal tangent basis provided by (11) $\mathbf{\Phi}_i$; $\nabla f(\mathbf{x})$ is equivalent to (8); its radius is ρ_i; and $\mathcal{U}_i$ is a set of points in the tangent space. Two things differentiate this from the original AtlasRRT algorithm. First, the size of a chart, also termed its validity region, is inversely proportional to the variance at the chart center, namely,

$$\rho_i \propto \mathbb{V}[f(\mathbf{x}_i)]^{\ 1}. \tag{14}$$

[g] There is also work reporting the use of a proprioceptive system.
[h] We provide technical details of our approach in Sec. 5.1.

Algorithm 1: GPAtlasRRT

$\mathcal{P} \leftarrow$ **GPAtlasRRT**($\mathcal{M}$, $\mathbb{V}_{\max}$)

input: A GP model, $\mathcal{M}$ and the set of parameters Ω, defining criteria to decide how to start, extend and end the exploration.

output: The best next action, $\mathcal{P}$, in the form of a path, if any, or $\varnothing$ otherwise.

1 $\mathbf{x_i} \leftarrow$ SELECTSURFACEPOINT($\mathcal{GP}$)
2 $\mathcal{C}_i \leftarrow$CREATECHART($\mathbf{x}_i$, $\mathcal{GP}$)
3 $\mathcal{A} \leftarrow$ ADDCHART($\mathcal{C}_i$)
4 **while** ISEXPANDABLE($\mathcal{A}$) **do**
5 $\quad \mathcal{C}_j \leftarrow$SELECTCHART($\mathcal{A}$)
6 $\quad \mathbf{x}_k \leftarrow$EXPANDCHART($\mathcal{C}_j$, $\mathcal{GP}$)
7 $\quad \mathcal{C}_k \leftarrow$ CREATECHART($\mathbf{x}_k$, $\mathcal{GP}$)
8 $\quad \mathcal{A} \leftarrow$ADDCHART($\mathcal{A}$, $\mathcal{C}_k$)
9 $\quad$ **if** $\mathbb{V}[f(\mathbf{x}_k)] > \mathbb{V}_{\max}$ **then**
10 $\quad\quad \mathcal{P} \leftarrow$ GETPATH($\mathcal{C}_i$, $\mathcal{C}_k$)
11 $\quad\quad$ **return** $\mathcal{P}$
12 **return** $\varnothing$

ρ_i is thus actually the radius of a ball centered at $\mathbf{x}_i$, whose intersection with the tangent space yields the disk shaped *chart*. The motivation behind this choice is that the more certain a point is to be on the surface, the larger the region of its chart on the predicted shape, whereas if more uncertainty is associated with the center, smaller exploratory steps will be preferred. Second, a number of points proportional to the size of the chart are sampled from a random uniform density on an annulus of the disk with internal and external radii 0.8ρ and ρ, respectively. The cardinality of this point set in the tangent space is proportional to its size, namely,

$$\#\mathcal{U}_i \propto \rho_i. \tag{15}$$

This implies that the larger the chart, the more samples are needed to obtain a good quantization of it.[i]

The first chart is the root node of an exploration tree (line 3). The question whether an atlas ISEXPANDABLE or not (line 4) is answered by checking whether there is at least one chart i with $\#\mathcal{U}_i \neq 0$. If the predicted surface is completely covered with charts, the while condition will fail and the algorithm terminates (line 12), otherwise it loops. The first step selects a chart to expand (SELECTCHART, line 5) from those charts i with a nonempty point set $\mathcal{U}_i \neq \emptyset$. Tree expansion is either depth first (selecting the most recently created chart), with probability $p = 0.4$, or a randomized

[i] Another advantage of working with a normalized and offset-free set, as mentioned in Sec. 3.1, is that the parameters that make the latter two expressions equal are tuned once, and remain fixed.

sample (across other charts), with probability $1 - p$. In the first iteration, $\mathcal{C}_i$ is the only chart, and thus guaranteed to be expandable.

Next, the EXPANDCHART operation, on the selected chart s, (line 6) chooses, from all the points j in its annulus $\mathbf{u}_{s,j} \in \mathcal{U}_s$, the point in the tangent plane $\mathbf{u}^*_{s,j}$ with the largest variance in the target function (when it is projected onto the object surface), that is,

$$\mathbf{u}^*_{s,j} = \arg\max_{\mathbf{u}_{s,j} \in \mathcal{U}} \mathbb{V}[f(\psi_s(\mathbf{u}_{s,j}))]. \tag{16}$$

In the Cartesian space, this new surface point is $\mathbf{x}_j = \psi_s(\mathbf{u}^*_{s,j})$. A chart, $\mathcal{C}_j$, is then created, centered on this point and added to the atlas (lines 7–8). When the new chart $\mathcal{C}_j$ is added, we must remove all the points in the annulus of every other chart $\mathcal{C}_i$ that correspond to surface points which are also covered by the disk $\mathcal{C}_j$.[j]

Finally, the expected variance of the center point of the most recent chart is compared against the input threshold $\mathbb{V}_{\max}$ (line 8). If the variance is less than the threshold, then atlas expansion continues. Otherwise, the best exploration path $\mathcal{P}$ is returned (lines 10–11). Recall that this path lies on the predicted surface. Thus, the controller to follow it must be compliant to avoid damage. The new tactile obser vations increase the training set, $\mathcal{S}^0$, reducing the uncertainty of the surface model. Figures 3 and 4 show the AtlasRRT in the process of expansion on the implicit

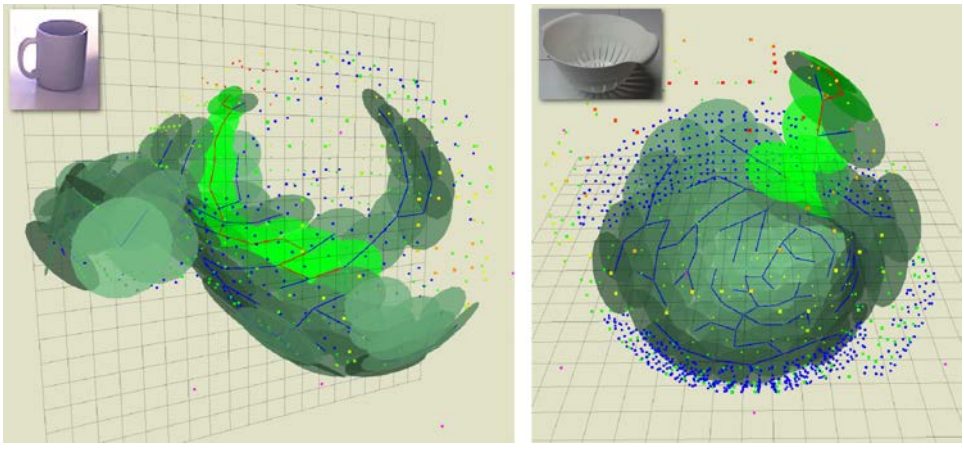


Fig. 3. The AtlasRRT expanding across the implicit surfaces of a mug (left) and a colander (right). The RRT used to create the atlas is marked in blue. The selected sequence of charts is highlighted in light green, and the associated path is marked in red (it is slightly obscured in the right panel). The robot tries to touch the object at the center of each chart in the sequence. Both objects are viewed from above.

[j]This was not mentioned in the first call of the ADDCHART in line 3, because at that point, there is only the root chart in the atlas.

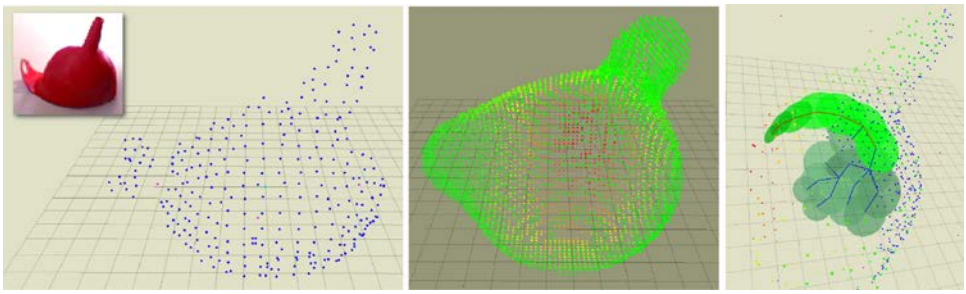


Fig. 4. A funnel (left-upper corner) is first seen by a depth camera. The segmented 3D points are shown in blue in the left figure to form the training set $\mathcal{S}^0$. The predicted shape by the GP on this set is shown in the middle obtained via a marching cube sampling algorithm. However, the GPAtlasRRT strategy does not require the explicit form of the predicted surface, as shown in the right figure. It works with the implicit form to devise the next best tactile exploration shown in brighter green.

surfaces for two partially explored objects. The next section describes how the GPAtlasRRT strategy is embedded in a tactile exploration scenario.

4.1. *Tactile exploration using GPAtlasRRT*

The GPAtlasRRT algorithm is the planner that drives tactile exploration. This must be embedded in an overall execution and inference loop (Algorithm 2). The initial, incomplete observation may be visual or tactile (lines to compute the model 2–5). Given this, the robot plans a sequence of touches, $\mathcal{P}$, (line 7), executes the next touch and updates the GPIS model. We have now presented the inference and planning components. Now, we present the execution component.

Several facts determine the difficulty of the problem and the shape of the solution for execution. First, to enable autonomous acquisition of near complete object models, the robot should ideally be able to reorientate the object to expose different surfaces. This requires that the object should be grasped by a second manipulator. Second, to minimize data errors due to object movement during exploration, the object must not move, so the grasp should be firm. However, a second feature of the problem is that the shape is not known. This makes obtaining a firm grasp chal lenging. We employed an underactuated manipulator (the Pisa/IIT soft hand[43]), which is both powerful and copes well with unmodeled contacts.

There is then, however, a third problem. The view of the grasped object contains not only the object but also the hand. So, the hand and object must be segmented from one another. Since underactuated hands do not typically possess position encoders, recovering the hand pose so as to segment the hand from the point cloud of the grasped object is nontrivial (line 3).

To tackle this, we sensorized the soft hand using inertial measurement units (IMUs)[44] to recover the hand configuration. Using this information, together with the arm configuration, we crop the scene point cloud to separate the points on the

Algorithm 2: Surface modeling via GPAtlasRRT

TactileExploration($\mathcal{Z}, \mathbb{V}_{\max}$)

input: An initial point cloud of the scene, $\mathcal{Z}$, and the desired variance, $\mathbb{V}_{\max}$, for the overall surface prediction.

output: The object model as a GP, $\mathcal{GP}$.

```
1  if isEmpty(Z) then
2      S^0 ← naiveProbe()
3  else
4      S^0 ← segmentObject(Z)
5  S ← generateTrainSet(S^0) GP ←computeModel(S)
6  while true do
7      P ←GPAtlasRRT(GP, V_max)
8      if P ≠ ∅ then
9          ApproachTo(P, GP)
10         S^0+ ←probeObject(P)
11         S ← updateTrainSet(S, S^0+)
12         GP ←computeModel(S)
13         MoveAway(GP)
14     else
15         return GP
```

object from those on the hand. The partial coverage of the object surface by the grasping hand limits the extent of the tactile exploration. This would require a re grasp maneuvre, which falls out of the scope of this paper.

Having grasped the object, segmented it, inferred the initial GP model, and planned a sequence of touches, the robot finger moves to a pose from which it can initiate a movement to make the first touch (approachTo). This must be a safe distance from the predicted surface and normal to the target contact point. Since the object shape representation captures its uncertainty, a coarse point cloud is com puted from the GP model and used to build a probabilistic collision map. Then, the robot moves to contact the surfaceprobeObject, resulting in contact or noncontact. There are two schemes, in one the probe touches each point in the path $\mathcal{P}$. In the other, only the final, high variance, point in $\mathcal{P}$ is touched.

The resulting position(s) $\mathbf{x}_i$ and target(s) $\mathbf{y}_i$ form a (series of) tuple(s) collected in the observation set $\mathcal{S}^{0+}$ (line 10). During touch motions, collision avoidance is dis abled and the probe moves compliantly. Once the end of the path is reached, the training set is updated (line 11). This is used to recompute a better model of the object surface, $\mathcal{GP}$ (line 12). We then move the probe away (line 13). When the $\mathcal{GP}$ model has a maximum variance of $\mathbb{V}_{\max}$ for any point on the implicit surface, that is, $\mathbb{V}[f(\mathbf{x})] < \mathbb{V}_{\max}, \forall \mathbf{x} \in \mathcal{X}$, exploration terminates.

The next section presents two experimental studies, one in simulation, where PROBEOBJECT is performed by raycasting on object meshes, and a real robot experiment.

5. Experimental Validation

To validate the approach, we first devised a simulation, where we performed tests on nine everyday objects, represented as polygonal meshes of Fig. 5. These were obtained using an RGBD sensor and a turntable. They are used as ground truth and to create simulated depth images. Each trial iterated the GPATLASRRT algorithm, Algorithm 1 in Sec. 4, until the shape was predicted with desired variance $\mathbb{V}_{\max} = 0.1$. The selected tactile actions were simulated using raycasting. Rays are uniquely defined by a chart center, as pivot point, and its normal, as direction. Thus, we can define ray–mesh intersections as touches, and nonintersections as points outside the surface. We adopted three different tactile schemes, thus forming three conditions, for a total of 27 full shape reconstructions. These are as follows:

- **Random Touch** for the first condition: the robot just attempted to touch a random point on the GP manifold. This repeated until the reconstructed shape had a maximum variance of 0.1. This was our control condition or baseline. Test results are shown in Table 1.
- **Single Poke** for the second condition: we used the GPATLASRRT by poking the last chart in the path. The convergence criterion was the same as for the random condition. Results in Table 1 show a significant reduction in the number of tactile actions required to reach the requested shape uncertainty.

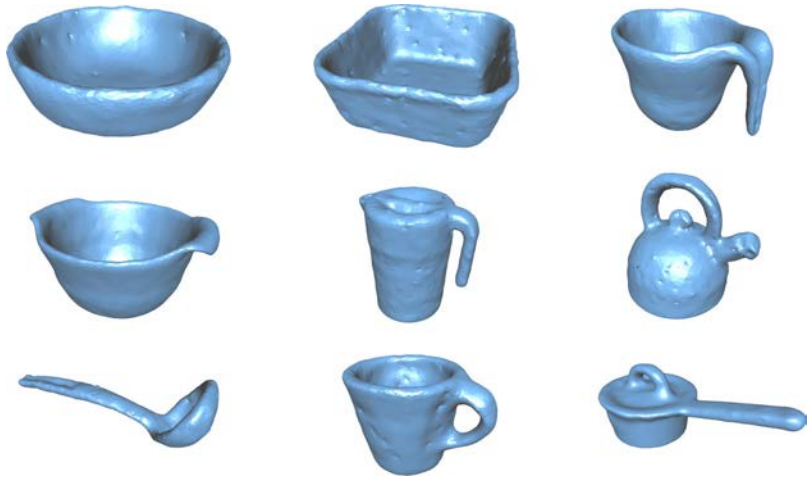


Fig. 5. Object meshes used as ground truth for our simulated tests. From top to bottom and from left to right: bowl A, bowl B, container A, container B, jug, kettle, spoon, mug and pot. The typical maximum dimension of each object is from 20 cm to 40 cm.

Table 1. Simulated results for all three conditions in terms of the required number of actions (steps) and the root-mean-squared error (RMSE) between the predicted shape and the ground truth mesh. RMSE is in m.

	Random poke		Single poke		Sliding	
Object	Steps	RMSE	Steps	RMSE	Steps	RMSE
Bowl A	67	0.0025	27	0.0023	8	0.0015
Bowl B	38	0.0038	18	0.0036	5	0.0028
Container A	124	0.0033	20	0.0035	11	0.0028
Container B	68	0.0062	19	0.0043	8	0.0026
Jug	106	0.0027	20	0.003	9	0.0025
Kettle	98	0.0031	17	0.0032	9	0.0029
Spoon	35	0.0058	10	0.0055	8	0.0031
Mug	238	0.0017	28	0.0020	12	0.0018
Pot	33	0.0035	12	0.0032	6	0.0028
Mean	**∼90**	**0.0036**	**∼19**	**0.0034**	**∼8**	**0.0025**

- **Sliding Touch** for the final condition: we used the full path generated by GPAtlasRRT. Starting from the root chart, we made simulated touches and from each one, re interpolated the path toward the next chart. This was repeated until the tip of the atlas branch was reached. As the virtual probe moves across each chart, many data points were gathered, in contrast to the single poke or random conditions. We hypothesized that this condition would be the best performer in terms of the quality of the reconstructed shape, and in terms of the number of charts traversed. Table 1 shows this to be correct.

The three experiments clearly show the superiority of the single poke and sliding touch methods in terms of number of required steps and in terms of quality of the produced mesh. We performed Mann–Whitney tests to find the statistical signifi cance of the difference between each pair of algorithms, by ranking their perfor mances.[k] For the number of separate touches until convergence, all pairs of algorithms were significantly different at $p < 0.001$ for a two tailed test. For the quality of implicit surface estimation, the difference between the sliding condition and the single poke was significant at $p < 0.05$ for a two tailed test.

As a final benchmark, Table 2 summarizes the comparison between the test methods and Fig. 6 shows some of the GPAtlasRRT sliding touch reconstructed shapes with the ground truth meshes next to them.[l]

5.1. *Robot experiments*

This section provides an empirical evaluation of Algorithm 2 on our real robotic platforms. As mentioned in Sec. 4.1, we do not implement a regrasping maneuvre to

[k]Despite not exploiting the paired nature of the data, this is a good test to use in the instance, as it avoids any assumptions about the underlying distribution of scores.
[l]Additionally, we recorded videos of the shape reconstructions, see `goo.gl/4GKYTp`.

Table 2. Overall comparison: GPAtlasRRT with sliding touch outperforms in terms of efficiency and accuracy. RMSE is in m.

Tests	Mean steps	Mean RMSE
Random touch	90	0.0036
Single poking	19	0.0034
Sliding touch	8	0.0025

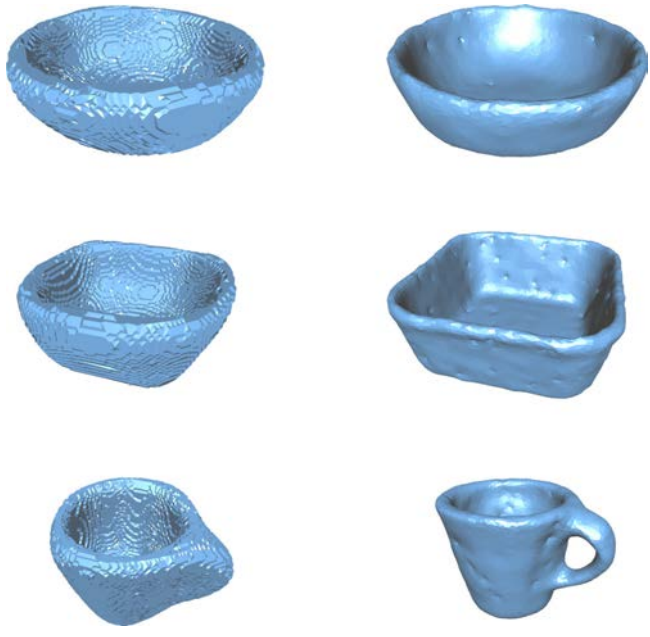


Fig. 6. Comparison of reconstructed shapes (left) with the ground truth meshes (right), obtained with our GPAtlasRRT via the sliding touch method.

overcome hand induced occlusions, and thus reconstructions in the real robot case are incomplete, as we segment the grasped part of the object.[m] The technical details are now given.

In this scenario, we employ our Vito and Boris robots. These are bimanual robots equipped with two KUKA LWR 4+, one Pisa/IIT soft hand[43] as one end effector, and the intrinsic tactile sensor configuration as introduced in Ref. 46. With Vito, we start a trial by handing the robot an object. Afterwards, the object is segmented with the help of the recently developed IMU based glove by Santaera *et al.*[44] to measure the hand configuration, and we remove the entire robot body from the scene. Other

[m]The implementation is mixed open-source `github.com/CentroEPiaggio/pacman-DR54`, heavily based on the Robot Operating System.[45] The GPAtlasRRT (Algorithm 1) is a submodule `github.com/pacman-project/gaussian-object-modelling`. As with the simulated results, we present an accompanying video `https://goo.gl/4GKYTp`.

typical filters such as pass through and downsampling were applied to speed up the overall pipeline. The acquired cloud contained an incomplete view of the object and constituted the initial training data for the GP, namely $\mathcal{S}^0$. Figure 8 shows the initial model. Then, a sequence of touches was performed. The experiments were performed using the single poke condition.[n] Planning for the bimanual and unimanual setups used MoveIt. We performed both IK solving and path planning using this, and rejected tactile touches with unfeasible paths. In the event of path planning failure, we simply restart the tactile exploration procedure.

The experimental results on Vito have shown that the grasping hand necessarily prevents full completion of the model, so an additional terminating condition is used.[o] On Boris, the object is not handed to the robot, but held by a clamp. Thus, we only used Boris's arm with the intrinsic tactile sensor. This choice was made so that — due to kinematic restrictions of this robot when operating bimanually without re grasp — the robot can reach and touch as large a proportion of the object's surface as possible, thus giving the most complete run of the tactile explo ration algorithm. Figure 7 shows the setup.

On Boris, we ran the tactile exploration algorithm on a white plastic jug. Figure 9 shows the evolution of the estimated model against the number of touches. The color of the points encodes the variance in the surface estimate, ranging from red (high variance) to blue (low variance). Figure 9(a) presents the initial model obtained from the point cloud. From left to right, the models are generated after, respectively, 5(b), 15(c), 25(d) and 32(e) touches. It is interesting to see that even after 15 touches, the model is already close to the final shape estimate for the jug, but the GP is uncertain

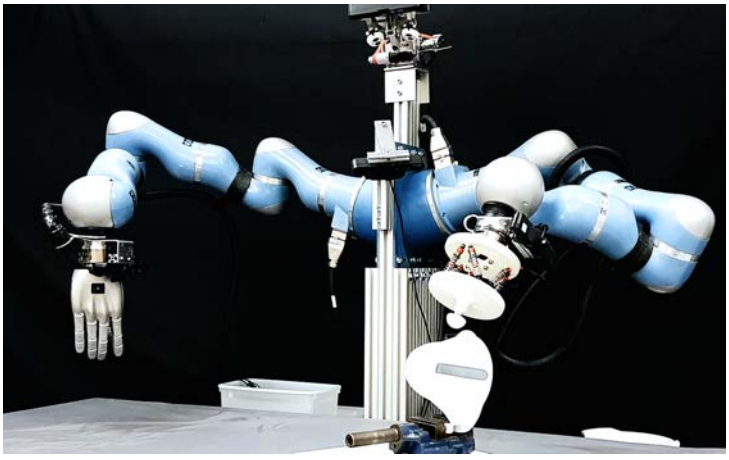


Fig. 7. Real experimental setup. Boris explores an object fixed on the table.

[n] The sliding condition requires more sophisticated impedance control than we had readily available. This makes sliding with our method a good piece of future work.
[o] This is a threshold for a number of failed consecutive attempts to execute a touch, and models the fact that it is not possible to touch areas occluded by the hand.

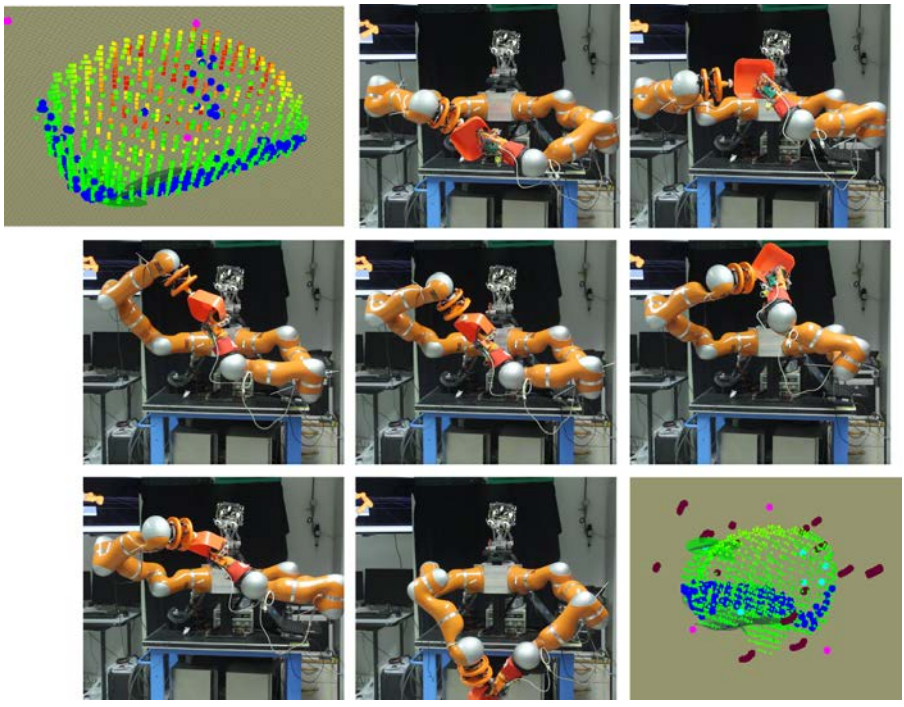
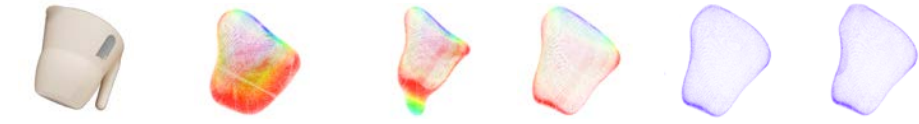


Fig. 8. Our Vito robot performs a tactile exploration action using the proposed GPAtlasRRT strategy. The per point color code is the same as in Fig. 1.



object	(a) from vision	(b) 5 t	(c) 15 t	(d) 25 t	(e) 32 t

Fig. 9. Experiment on Boris. Left, the object and the initial GP model from the single view from the depth camera. The model improvement as the number of touches increases from left to right. The colors represent the variance from red (high variance) to blue (low variance). After 15 touches, the model already converges to the shape of the jug. The handle is excluded from inference since it is grasped and thus not explored. The GPAtlasRRT algorithm terminates after 32 touches.

and so the procedure keeps exploring until the variance is reduced below the threshold everywhere.

6. Conclusions and Future Work

This paper presented a method for exploring an object on all sides with a finger. Key to our approach was a combined shape representation and planning algorithm

(GPAtlasRRT). The planner allows the robot to estimate the location of unseen and untouched surface. Together they allow planning of local tactile exploration of an incompletely modeled object. We demonstrated the benefits both in simulation, and on a real robot. The real robot system also demonstrated the ability to grasp an object with one hand, segment this hand from the object in an initial point cloud, and then extend the model with touches guided by GPAtlasRRT.

This local tactile planning strategy required bringing together GPISs and the determination of implicitly defined manifolds via continuation techniques. This exploited the ability of GPs to naturally represent model uncertainty.

The beneficial features of the approach are several. First, the planning method does not require the computation of the explicit form of the entire predicted shape. Second, the strategy makes no assumptions about the exploratory probe. Third, it can plan sequences of tactile actions across a contiguous portion of the object surface, thus providing a detailed surface reconstruction. Fourth, the robot implementation allows the robot to explore an object as it holds it.

The proposed strategy was compared to a naive one, where touch rays were directed randomly. Our strategy outperforms this, whether using a single touch, or a touch sequence. The strategy was also tested successfully using our Vito and Boris robots. The previously published methods simplified the setting by placing the object on a table and then planning actions in a Cartesian space. This does not permit the robot to plan to traverse to the unseen back surfaces of objects with a single finger, or to explore objects rotated as they held in the hand. Instead, by creating an object centered representation, the method presented here is able to handle these cases, which are the characteristics of human tactile exploration of objects.

Several points deserve further attention and future work. Perhaps most relevant to this work is the consideration of gradient observations as described in Ref. 47, especially due to our hardware setup. This feature has been presented in related work but not previously exploited. Another interesting point arises when discussing how to locally explore the model, that is, the direction to move within a chart. A third interesting topic is the use of the proposed strategy to drive a control loop where the controller command can be part, or even just the first single step, of the exploratory path. According to our experience, this is a feasible and promising road to explore. Implementation issues also lead to nontrivial scientific problems. For example, the problem of maintaining a stable grasp during bimanual exploration needs to be more thoroughly addressed. We found that an underactuated hand could maintain a firm hold of the object, but movements of the object in hand could seriously degrade the model quality. This could be addressed by re estimating the object position in hand by best fitting its pose against the parts of the model that are most certain. The problem can also be addressed by carefully controlling the applied forces during exploration. We believe that position based planners are inadequate for this task, and that various compliant/force control strategies could be applied.

Acknowledgments

The authors would like to thank E. Farnioli for the fruitful discussions on GPs, as well as to G. Santaera for the support with the IMU based glove. We gratefully acknowledge the receipt of European Commission Grant PacMan EC FP7 ICT 600918.

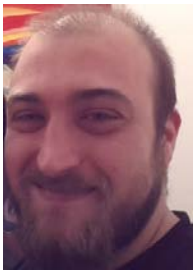
References

1. D. Kragic and H. I. Christensen, *Survey on Visual Servoing for Manipulation*, Technical Report, ISRN KTH/NA/P 02/01 SE (Royal Institute of Technology, Stockholm, Swe den, 2002).
2. J. Nunez Varela and J. L. Wyatt, Models of gaze control for manipulation tasks, *ACM Trans. Appl. Percept.* **10**(4) (2013) 20.
3. E. Arruda, J. L. Wyatt and M. Kopicki, Active vision for dexterous grasping of novel objects, in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems* (*IROS*) (IEEE, Daejeon, 2016), pp. 2881 2888.
4. M. Kopicki, R. Detry, M. Adjigble, A. Leonardis and J. L. Wyatt, One shot learning and generation of dexterous grasps for novel objects, *Int. J. Robot. Res.* **35**(8) (2015) 959 976.
5. G. Kootstra, M. Popovi, J. Jørgensen, K. Kuklinski, K. Miatliuk, D. Kragic and N. Kruger, Enabling grasping of unknown objects through a synergistic use of edge and surface information, *Int. J. Robot. Res.* **34** (2012) 26 42.
6. R. J. Johansson and K. J. Cole, Sensory motor coordination during grasping and ma nipulative actions, *Curr. Opin. Neurobiol.* **2** (1992) 815 823.
7. C. Zito, M. Kopicki, C. Borst, F. Schmidt, M. Roa and J. L. Wyatt, Sequential trajectory re planning with tactile information gain for dexterous grasping under object pose un certainty, in *2013 IEEE/RSJ Int. Conf. Intelligent Robots and Systems* (*IROS*) (IEEE, Tokyo, 2013), pp. 4013 4020.
8. L. P. Jentoft, Q. Wan and R. D. Howe, Limits to compliance and the role of tactile sensing in grasping, in *IEEE Int. Conf. Robotics and Automation* (IEEE, Hong Kong, 2014), pp. 6394 6399.
9. M. Bjorkman, Y. Bekiroglu, V. Hogman and D. Kragic, Enhancing visual perception of shape through tactile glances, in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems* (IEEE, Tokyo, 2013), pp. 3180 3186.
10. P. Hebert, T. Howard, N. Hudson, J. Ma, J. Burdickand and W. Joel, The next best touch for model based localization, in *IEEE Int. Conf. Robotics and Automation* (IEEE, Karlsruhe, 2013), pp. 99 106.
11. A. Petrovskaya and O. Khatib, Global localization of objects via touch, *IEEE Trans. Robot.* **27**(3) (2011) 569 585.
12. H. Gu, S. Fan, H. Zong, M. Jin and H. Liu, Haptic perception of unknown object by robot hand: Exploration strategy and recognition approach, *Int. J. Hum. Robot.* **13**(3) (2016) 1650008.
13. H. Gu, Y. Zhang, S. Fan, M. Jin and H. Liu, Grasp configurations optimization of dexterous robotic hand based on haptic exploration information, *Int. J. Hum. Robot.* **14**(4) (2017) 1750013.
14. R. Bajcsy, Active perception, *Proc. IEEE* **76**(8) (1988) 966 1005.
15. W. E. L. Grimson and T. Lozano Perez, Model based recognition and localization from sparse range or tactile data, *J. Robot. Res.* **3**(3) (1984) 3 35.

16. O. Faugeras and M. Hebert, A 3D recognition and positioning algorithm using geomet rical matching between primitive surfaces, in *Proc. 8th Int. Joint Conf. Artificial Intelli gence* (William Kaufmann, Karlsruhe, 1983), pp. 996 1002.
17. S. Shekhar, O. Khatib and M. Shimojo, Sensor fusion and object localization, in *IEEE Int. Conf. Robotics and Automation* (IEEE, San Francisco, 1986), pp. 1623 1628.
18. R. Bajcsy, S. Lederman and R. L. Klatzky, *Machine Systems for Exploration and Manipulation: A Conceptual Framework and Method of Evaluation*, Technical Report, MS CIS 89 03 (Moore School of Electrical Engineering, Philadelphia Department of Computer and Information Sciences, 1989).
19. S. Dragiev, M. Toussaint and M. Gienger, Gaussian process implicit surfaces for shape estimation and grasping, in *Proc. IEEE Int. Conf. Robotics and Automation* (IEEE, Shanghai, 2011), pp. 2845 2850.
20. N. Sommer, M. Li and A. Billard, Bimanual compliant tactile exploration for grasping unknown objects, in *IEEE Int. Conf. Robotics and Automation* (IEEE, Hong Kong, 2014), pp. 6400 6407.
21. C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning* (MIT Press, Boston, 2006).
22. L. Jaillet and J. M. Porta, Path planning under kinematic constraints by rapidly ex ploring manifolds, *IEEE Trans. Robot.* **29**(1) (2013) 105 117.
23. P. K. Allen and R. Bajcsy, Robotic object recognition using vision and touch, in *Proc. 9th Int. Joint Conf. Artificial Intelligence* (Morgan Kaufmann, Milan, 1987), pp. 1131 1137.
24. P. K. Allen and P. Michelman, Acquisition and interpretation of 3D sensor data from touch, *IEEE Trans. Robot. Autom.* **6**(4) (1990) 397 404.
25. K. Roberts, Robot active touch exploration: Constraints and strategies, in *IEEE Int. Conf. Robotics and Automation* (IEEE, Cincinnati, 1990), pp. 980 985.
26. S. Caselli, C. Magnanini, F. Zanichelli and E. Caraffi, Efficient exploration and recogni tion of convex objects based on haptic perception, in *IEEE Int. Conf. Robotics and Automation* (IEEE, Minneapolis, 1996), pp. 3508 3513.
27. M. Moll and M. A. Erdmann, Reconstructing the shape and motion of unknown objects with active tactile sensors, in *Springer Tracts in Advanced Robotics*, Chap. 17 (Springer Verlag, Berlin, 2003), pp. 293 310.
28. M. Meier, M. Schopfer, R. Haschke and H. Ritter, A probabilistic approach to tactile shape reconstruction, *IEEE Trans. Robot.* **27**(3) (2011) 630 635.
29. S. Dragiev, M. Toussaint and M. Gienger, Uncertainty aware grasping and tactile explo ration, in *IEEE Int. Conf. Robotics and Automation* (IEEE, Karlsruhe, 2013), pp. 113 119.
30. S. Ottenhaus, M. Miller, D. Schiebener, N. Vahrenkamp and T. Asfour, Local implicit surface estimation for haptic exploration, in *IEEE Humanoids* (IEEE, Cancun, 2016), pp. 850 856.
31. A. Bierbaum, M. Rambow, T. Asfour and R. Dillmann, A potential field approach to dexterous tactile exploration of unknown objects, in *8th IEEE RAS Int. Conf. Humanoid Robots* (IEEE, Daejeon, 2008), pp. 360 366.
32. O. Williams and A. Fitzgibbon, Gaussian process implicit surfaces, in *PASCAL Pattern Analysis, Statistical Modelling and Computational Learning Gaussian Processes in Practice Workshop* (2007).
33. T. Matsubara, K. Shibata and K. Sugimoto, Active touch point selection with travel cost in tactile exploration for fast shape estimation of unknown objects, in *IEEE Int. Conf. Advanced Intelligent Mechatronics* (*AIM*) (IEEE, Banff, 2016), pp. 1115 1120.
34. U. Martinez Hernandez, T. J. Dodd, M. H. Evans, T. J. Prescott and N. F. Lepora, Active sensorimotor control for tactile exploration, *Robot. Auton. Syst.* **87** (2017) 15 27.

35. N. Jamali, C. Ciliberto, L. Rosasco and L. Natale, Active perception: Building objects' models using tactile exploration, in *IEEE RAS 16th Int. Conf. Humanoid Robots* (*Humanoids*) (IEEE, Cancun, 2016), pp. 179 185.
36. N. Tosi, O. David and H. Bruyninckx, Action selection for touch based localisation trading off information gain and execution time, in *IEEE Int. Conf. Robotics and Automation* (*ICRA 2014*) (IEEE, Hong Kong, 2014), pp. 2270 2275.
37. N. F. Lepora, K. Aquilina and L. Cramphorn, Exploratory tactile servoing with active touch, *IEEE Robot. Autom. Lett.* **2** (2017) 1156 1163.
38. M. E. Henderson, *Computing Implicitly Defined Surfaces: Two Parameter Continuation*, Technical Report 18777 (T. J. Watson Research Center, IBM Research Division, 1993).
39. S. M. LaValle, Motion planning, *IEEE Robot. Autom. Magn.* **18**(1) (2011) 79 89.
40. M. Li, K. Hang, D. Kragic and A. Billard, Dexterous grasping under shape uncertainty, *Robot. Auton. Syst.* **75** (2016) 352 364.
41. J. Porta, L. Ros, O. Bohigas, M. Manubens, C. Rosales and L. Jaillet, The CUIK suite: Analyzing the motion closed chain multibody systems, *IEEE Robot. Autom. Magn.* **21**(3) (2014) 105 114.
42. A. Bicchi, J. K. Salisbury and D. L. Brock, Contact sensing from force measurements, *Int. J. Robot. Res.* **12**(3) (1993) 249 262.
43. M. G. Catalano, G. Grioli, E. Farnioli, A. Serio, C. Piazza and A. Bicchi, Adaptive synergies for the design and control of the Pisa/IIT soft hand, *Int. J. Robot. Res.* **33**(5) (2014) 768 782.
44. G. Santaera, E. Luberto, A. Serio, M. Gabiccini and A. Bicchi, Low cost, fast and ac curate reconstruction of robotic and human postures via IMU measurements, in *IEEE Int. Conf. Robotics and Automation* (IEEE, Stockholm, 2015), pp. 2728 2735.
45. M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler and A. Y. Ng, ROS: An open source robot operating system, in *ICRA Workshop on Open Source Software* (Willow Garage, Kobe, 2009).
46. C. Rosales, A. Ajoudani, M. Gabiccini and A. Bicchi, Active gathering of frictional properties from objects, in *2014 IEEE/RSJ Int. Conf. Intelligent Robots and Systems* (IEEE, Chicago, 2014), pp. 3982 3987.
47. E. Solak, R. Murray Smith, W. E. Leithead, D. J. Leith and C. E. Rasmussen, Derivative observations in Gaussian process models of dynamic systems, in *Advances in Neural Information Processing Systems*, eds. S. Becker, S. Thrun and K. Obermayer (Curran Associates, Vancouver, 2003), pp. 1057 1064.

Carlos J. Rosales received his M.Sc. and Ph.D. degrees from the Universitat Politècnica de Catalunya, Spain, in 2006 and 2013, respectively. From 2013–2016, he was a Research Fellow at the Research Center E. Piaggio (Università di Pisa). He is Founder and CTO of Beta Robots, Barcelona. His research concerns kinematic and dynamic limitations for grasp planning under uncertainty, trajectory planning, and robot motion generation.



Federico Spinelli gained his Bachelor's in Computer Software Engineering and his M.Sc. in Mechatronics, Robotics and Auto mation Engineering from the Università di Pisa in 2010 and 2014, respectively. From 2014–2016, he was a Research Fellow at the Research Center E. Piaggio (Università di Pisa) on the PaCMan project, specializing in 3D vision for robot manipulation. He is a Robotics Engineer at QDesign Srl.

Marco Gabiccini received the Laurea degree (*cum laude*) and the Ph.D. both from the University of Pisa, Pisa, Italy, in 2000 and 2006, respectively. While doing his Ph.D., he was a Visiting Scholar at the GearLab, The Ohio State University, Columbus, USA, from 2003–2004. Since 2001, he has been doing research at the Department of Mechanical, Nuclear and Production Engineering, University of Pisa. In 2006, he joined the Interde partmental Research Center "E. Piaggio" and in 2011 also joined the Advanced Robotics Department in the Italian Institute of Technology (IIT), Genova, Italy.

He is currently an Associate Professor in the Department of Civil and Industrial Engineering (DICI).

He teaches robotics, applied mechanics and vehicle dynamics at the University of Pisa, School of Engineering. His main research interests are in the field of theory of gearing, vehicle dynamics, geometrical methods in robotics and in the areas of dynamics, kinematics and control of complex mechanical systems.

Claudio Zito gained his BA in Mathematics and Computer Science from the Università di Siena (2006) and his M.Sc. in Computer Science from the Università di Pisa (2010). He obtained his Ph.D. in AI and Robotics from the University of Birmingham in 2016. His main research interests concern robot grasping, robot manipulation, artificial intelligence and machine learning. He is currently a Lecturer in Intelligent Robotics in the School of Com puter Science at the University of Birmingham.

Jeremy L. Wyatt is a Professor of Robotics and Artificial In telligence at the University of Birmingham. He has a BA in Theology from the University of Bristol (1990), an M.Sc. in Artificial Intelligence from the University of Sussex (1992), and a Ph.D. in Machine Learning from the University of Edinburgh (1997). He has (co)authored more than 100 refereed papers, edited three books, received two best paper awards, and super vised a BCS distinguished doctoral dissertation winner. He was a Project Coordinator for two European FP7 projects, CogX and PaCMan, and par ticipated in three more (CoSy, GeRT and Strands). He was a Leverhulme Fellow (2006–2008) and a Seelye Fellow (2018). He works on robot task planning, robot motion planning, robot manipulation, robot vision and robot learning in open, un certain and unfamiliar environments.