Autonomous Robots manuscript No.
(will be inserted by the editor)

GPAtlasRRT: A tactile exploration strategy for novel object shape modeling

Carlos J. Rosales · Claudio Zito · Federico Spinelli · Marco Gabiccini · Jeremy L. Wyatt

Received: date / Accepted: date

Abstract Information on object shape is a fundamental parameter for robot-object interaction to be successful. However, incomplete perception and noisy measurements impede obtaining accurate models of novel objects. Especially when vision and touch simultaneously are envisioned for learning object models, a representation able to incorporate prior shape knowledge and heterogeneous uncertain sensor feedback is paramount to fusing them in a coherent way. Moreover, by embedding a notion of uncertainty in shape representation allows to more effectively bias the active search for new tactile cues. In this work, we use a Gaussian Process (GP) as such representation. Then, using the fact that the 0-levelset of the GP — the surface of the object — is an implicitly defined manifold, we borrow the AtlasRRT algorithm concept to simultaneously: (i) build an atlas via continuation methods that locally parameterizes the object and that is used to select the next-best touch, and (ii) use an RRT-like strategy to devise directions for the expansion of the atlas to trade completeness in the exploration for efficiency. We integrate this strategy in a framework for iterative learning the object shape and implement a tactile exploration scenario in the bi-manual setting of our Vito robot. The experimental results confirm that the devised strategy yields better models compared to a naive strategy, and does it with a reduced number of touches.

Keywords Next-best touch · Shape modeling · Tactile exploration

Fig. 1 Our Vito robot performing a tactile exploration on an object (left). The current touch is suggested by the GPAtlasRRT strategy (right).

1 Introduction

One of the main reasons that makes autonomous grasping tasks, an ubiquitous and fundamental task in robotics, very challenging is that object properties required for grasp planning like shape and friction are not known in advance. This requires robots to perceive objects around them based on sensory information. However, sensory systems are prone to errors. As an example, when considering vision, some sources of noise are imperfect scene segmentation, occlusions, and poor lighting conditions.

Although for robotic grasp planning the use of vision has been studied more in depth by Kragic and Christensen (2002), the superior capabilities of humans in interacting with the environment come from rich sensory information where both visual and haptic modalities contribute to the combined percept. Working toward this ability in robots, the goal of our work is to complement visual information with tactile sensing in order to acquire 3D object models.

This work is supported by EC-FP7-ICT-600918, PacMan.

C. J. Rosales, F. Spinelli, and M. Gabiccini
Centro di Ricerca "E. Piaggio", Univ. di Pisa, Italy.
E-mail: carlos.rosales@for.unipi.it

C. Zito and J. L. Wyatt
Intelligent Robotics Laboratory, Univ. of Birmingham, UK.

Even if tactile perception needs have been authoritatively spelled out by Bajcsy (1988) in the 1980's and early perception algorithms date back to the same years (Grimson and Lozano-Perez 1984; Faugeras and Hebert 1983; Shekhar et al 1986; Bajcsy et al 1989), touch-based perception has lagged behind vision for two main reasons. The first is technological: standardized touch sensors are not easily available and have to be hand crafted for the specific robot and task. The second is intrinsic to the perception modality which requires the mechanical interaction of the sensor with the object being perceived with its inevitable perturbation and, to minimize this effect, calls for a complex control of the ongoing movement of the sensor: a requirement which is completely absent, e.g., in vision.

The scenario we pursue is motivated by tactile exploration when other sensor modalities, like vision, already provided initial and incomplete information on object shape. More in details, we provide a systematic methodology to plan the next-best tactile exploration action. This is intrinsically a contact hypothesis that needs to be accepted or discarded after execution. Despite being the hypothesis verified or falsified, it helps to improve the object shape prediction up to a pre-specified variability.

The object shape representation we employ is a probabilistic one and it is based on Gaussian Process (GP) (Rasmussen and Williams 2006), a machine learning formalism for nonlinear function regression that naturally provides a quantification of multi-modal uncertainty about object point estimates, i.e. the variance of the estimate. Being the 0-levelset of an *unknown* function an implicitly defined surface that represents the estimated outer shape of an object, it comes handy to interpret it as a manifold and build on recent results on sample-based exploration of general manifolds (Jaillet and Porta 2013), and implement similar algorithms to define hypotheses on where to sample next — next-best touch — to reduce uncertainty. More in details, without the need to embed the manifold in its ambient space, we build a collection of charts (atlas) that locally parameterizes it, and select among the points on the current atlas the one(s) fulfilling the search criteria: this represents the location where the next touch will be directed. Then, the growth of the atlas itself does not follow a predefined sequence of steps, but an RRT-like strategy is employed to devise random directions to expand the atlas. We propose a search criteria that looks for points on the estimated surface with the variance of being there larger than a pre-specified threshold. From another point of view, we look in all for an estimated surface, such that, any arbitrary point on it has a variance smaller than the pre-specified threshold. The expansion process is then repeated iteratively after the execution of the suggested touch. It certainly trades completeness in the novel shape exploration for efficiency: the RRT drives the growth of the atlas toward regions of the object surface that are more uncertain, delaying the refinement of areas that have been already explored.

The ending condition is met when no candidate for the next best touch is found by the GPAtlasRRT algorithm, which means the that object shape prediction meets the requirement on variability. Naturally, the smaller the threshold (i.e. a more accurate model is required), the higher the number of touches. This is the only input to the devised strategy, given either by a higher level module or the user.

The structure of the paper is described in the following: in Sec. 2 we first review previous work related to tactile exploration and object shape representations. 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 early attempts to exploit active tactile exploration with passive stereo vision for object recognition was proposed by Allen and Bajcsy (1987). In that paper, a rigid finger-like tactile sensor was used to trace along the surface with predefined movement cycles and provided a limited amount of information on object surface. The work was later extended to develop different exploratory procedures to acquire and interpret 3D touch information Allen and Michelman (1990). The exploratory procedures were, however, commanded by a human experimenter and therefore not linked to a fully autonomous system.

Single finger tactile exploration strategies for recognizing polyhedral objects have also been presented and evaluated in simulation, see for instance the work by Roberts (1990) and Caselli et al (1996). In Moll and Erdmann (2003) a method for reconstructing shape and motion of an unknown convex object using three sensing fingers is presented. In that approach, friction properties must be known in advance and the surface is required to be smooth, i.e., it must have no corners or edges. Moreover, multiple simultaneous sensor contacts points are required resulting in additional geometric constraints for the setup.

In the work of Petrovskaya and Khatib (2011), exploratory procedures have been considered with the aim to globally localize an object of known shape. Since the Bayesian posterior estimation for objects in 6D is known to be computationally expensive, that paper proposes an efficient approach, termed Scaling Series, that approximates the posterior by particles. For fully constraining datasets, that approach performs the estimation in under 1 s with very high reliability.

In the paper presented by Meier et al (2011) the tactile shape reconstruction employs a Kalman filter, while in Bierbaum et al (2008) the tactile exploration is guided by Dynamic Potential Fields for motion guidance of the fingers. Here, the authors show that grasp affordances may be generated from geometric features extracted from the contact point set extracted during tactile exploration.

Interestingly, Sommer et al (2014) proposes a bimanual compliant tactile exploration that uses the GP representation to smooth noisy point data, but does not exploit the GP representation to define specific exploratory strategies.

Dragiev et al (2011) presents one of the first works that employs Gaussian Process Implicit Surfaces (GPIS) for the concurrent representation of the object shape and to guide grasping actions towards the object. However, that work concentrates only on the mean of the shape distribution, i.e. the maximum a posteriori (MAP) estimate of the shape and practically ignores one of the gains of the GP — the error bars. Later work by the same authors (Dragiev et al 2013) offers also a way to give preference to regions of the model with particular certainty level and introduce the notion of explore-grasp and exploit-grasp primitives.

Bjorkman et al (2013) focuses the attention on building object models that can be extracted with a small number of actions (multiple touches at once by tactile arrays) with the ultimate aim of understanding the category objects belong to, rather than exhaustively trying to explore the whole object. In that paper, the implicit function representation of the object surface is modelled by Gaussian Process regression as well, where the shape of the GP is governed by the thin plate covariance function derived by Williams and Fitzgibbon (2007). A set of predefined tactile glances are performed on the object: however, these are not updated as the object model gets refined as successive touches are performed. That paper, though, is the closest work to ours among the references. The main difference being: the space in which the next-best exploratory action is computed, the grain size to compute the predicted shape, and the ending condition for the overall algorithm. Bjorkman et al (2013) proposes that the the exploratory actions are to be searched in a discrete space in the vertical direction and the approach angle, which seem extrinsic to the shape model. Using the ambient space instead of the intrinsic representation does not guarantee that the new observations will be on the desired shape region to be explored. Moreover, since they are interested in a model that is useful for categorization, they propose Zernike moments to make it affine invariant, for which a fine-grained explicit representation is required. We also propose to compute the predicted shape but with a very coarse grain for collision avoidance purposes, an issue that is not considered in that work. Finally, the number of actions, or touches in that case, are limited to a certain number, and then ordered according to the closest point on the implicit function with higher variance. In contrast, we set the highest expected variance in the shape prediction, so we explore until, probabilistically speaking, that goal is achieved.

3 Problem statement

Shape is one fundamental object property used in several robotic tasks (Bajcsy et al 1989). But what makes a surface representation to be good? We guess the best answer could be: it depends on what you need to do with it. Nevertheless, there are several features that can be ticked or not accordingly, such as: accuracy, conciseness, intuitive parametrization, local support, affine invariant, arbitrary topology, guaranteed continuity, efficient rendering and efficient intersections, among others. There are some that are not really special for most robotic tasks such as intuitive parametrization or conciseness. Since we are dealing with arbitrary novel objects, probably the capacity to represent arbitrary topologies with guaranteed continuity is a good must. Certainly implicitly defined surfaces have these properties. There are several ways to define an implicitly defined surface, e.g. via algebraic equations, blobby models and variational surfaces. These classical specifications do not consider any uncertainty measure though. There is where the work by Williams and Fitzgibbon (2007) comes in, introducing Gaussian Process implicit surfaces (GPIS). This might not be the only representation to account for uncertainty, but it surely meets the requirements for a good shape representation in such robotic contexts, as it has been shown in previous works described in the preceding section. Subsection 3.1 is dedicated to introduce the notation for GPIS.

Such a representation considers the estimated surface to be the mean value of a Gaussian Process, that is the 0-level of an implicitly defined manifold. Henderson (1993) provides a way to compute implicitly defined surfaces via continuation techniques. That work has evolved in different manners for different purposes, including one that is of particular interest for our approach: the AtlasRRT algorithm, a path planning method for constrained mechanisms by Jaillet and Porta (2013), where a combination of continuation techniques with Rapidly-exploring Random Trees (RRT) (LaValle 2011), allows to compute paths in a constrained configuration space, i.e. a manifold embedded in an ambient space. However, as presented, there is no notion of uncertainty of the manifold itself —recall, this is the object surface in our case—, and along with the concept of GPIS, constitutes another key ingredient for our strategy. Subsection 3.2 describes the basic idea behind the AtlasRRT algorithm with some borrowed notation too.

The last, but not least, ingredient is the equipment. In Subsection 3, we enumerate considerations for the hardware that is to execute a tactile exploration, as well as possible limitations of that. Finally, with all these ingredients in mind, we formally define the problem we aim to solve in the last subsection.

3.1 Gaussian Process Implicit Surfaces

A surface in 3D 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 $F(\mathbf{x}, f(\mathbf{x})) = 0$, and the surface of our interest raises when setting $y = 0$ — the 0-level set. Naturally, the value of f (or y) increases to a positive value in outward normal direction to the surface (∇F), and decreases to a negative value in the inward direction ($-\nabla F$).

A Gaussian process (GP), on the other hand, "is a collection of random variables, any finite number of which have a joint Gaussian distribution" (Rasmussen and Williams 2006, Def. 2.1), and 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 $i = 1, \ldots, n$, i.e. points, its corresponding noise characterization σ_i and its target y_i, observed from a real process, one can in fact use the GP for regression for that training set, by specifying $m(\mathbf{x} = 0)$, and a covariance function according to the nature of the process, 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 with mean and variance arguments, respectively, $\mathcal{X} \in \mathcal{S}$ the subset corresponding to the input space of the training set, $K(\cdot, \cdot)$ the covariance matrix formed with elements $k_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, for all $i, j : \mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$, and $\boldsymbol{\sigma}$ the n-vector corresponding to the noise characterization of the ith observation, therefore, I is an identity matrix $n \times n$, and finally, $\mathbf{y}$ also the n-vector that collects the targets similarly. But the interesting part is the ability to predict a target, y_*, given a test input $\mathbf{x}_*$, where this expression can be block-expanded, respecting probability principles (Rasmussen and Williams 2006), to

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

where $\mathcal{X}_*$ is the set of test inputs, and $\mathbf{y}_*$ their corresponding predicted values by the trained GP model, arranged in the same n-vector fashion. For simplicity, we dropped the arguments of the matrices such that $K = K(\mathcal{X}, \mathcal{X})$, $K_* = K(\mathcal{X}, \mathcal{X}_*)$ and $K_{**} = K(\mathcal{X}_*, \mathcal{X}_*)$. Thus, the two predictive equations can be derived from algebraic manipulation to

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

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

The covariance function specifies a distribution over functions with a notion of nearness among them, and it is a critical ingredient to select the appropriate one to get coherent predictions out of a GP model, a problem sometimes referred to as the *kernel trick*. There is actually a full chapter dedicated to its study by Rasmussen and Williams (2006, see Ch. 4), however, none of them seem adequate for points belonging to an implicitly defined surface. Williams and Fitzgibbon (2007) comes with the idea to use the thin-plate concept from data interpolation, and obtained good results for a surface in 3D with the covariance function

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

with the change of variables $r = \|\mathbf{x} - \mathbf{x}'\|_2$ using the 2-norm as the distance between points, for clarity, and with R being the largest r that can be observed within the training set —recall that this is a closed pointset in 3D taken from the object surface observations. In that case, the training set is noise-free, so besides adopting the same covariance function for our strategy, we also extend the training set to be the composition of points on the surface, $\mathcal{S}^0$, with tuples of the form $s_i = (\mathbf{x}_i, \sigma_i, 0)$, points outside the surface, $\mathcal{S}^+$, with tuples of the form $s_i = (\mathbf{x}_i, 0, +1)$ and points inside the surface, $\mathcal{S}^-$, with tuples of the form $s_i = (\mathbf{x}_i, 0, -1)$. Thus, the training set $\mathcal{S} = \mathcal{S}^0 \cup \mathcal{S}^+ \cup \mathcal{S}^-$, and its cardinality is the sum of the respective cardinalities. Without loss of generality, but a slight gain in efficiency and parameter tuning, we can 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 by Li et al (2016). Of course, the model exploitation requires a re-scaling and re-centering processing step.

Up to this point, we are able to compute the expected predicted target $y_* = \bar{f}(\mathbf{x}_*)$ and the associated prediction variance $\mathbb{V}[f(\mathbf{x}_*)]$ for any given input in ambient space, namely $\mathbb{R}^3$ in our case, given a training set $\mathcal{S}$. This means that if we evaluate exhaustively a 3-dimensional box-grid containing the implicitly defined surface, the points on the surface are those where $y_* \simeq 0$ with an associated variance. This is all we need to explore and compute efficiently candidate points on the surface that improve the overall variance of the predicted shape, and not with a complete rendering as the box-grid evaluation, or even a marching cube algorithm, that requires time.

However, we must not forget that there is an actual probe going towards the candidate points, and probably the best direction to follow is that of the normal to the predicted surface at the candidate point, mostly to avoid local undesired collisions, so it would be handy to have also a prediction of that out of the same GP. In our case, the normal direction is actually parallel to the gradient of the function, a concept we will need in further sections as well. If we consider the posterior mean of the GP given in (4) for a single test point, we have

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

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 computed as

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

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

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

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

3.2 Atlas of an implicitly-defined manifold

In general, it is not possible to obtain always a global parametrization of an implicitly defined manifold embedded in an ambient space. In our case, the manifold of attention is to be the surface of an object. Henderson (1993) provides a precise method to compute the surface via local parametrization called *disks*. The concept is that there a disk contains an exponential mapping, explicitly represented by a tangent basis at a domain point on the surface, and the size of the disks corresponds to validity of that mapping. The disks are further projected onto the surface using a logarithmic mapping. The method typically starts with an initial point, $\mathbf{x} \in \mathbb{R}^3$, known to be on the surface, $f(\mathbf{x}) = 0$ (or very closed and projected onto it), and creates the first unbounded disk. From there, new disks are created, and carefully intersected with the previous ones to avoid getting into the same areas already covered, to create new disks and bound others. The method continues until there are no unbounded disks. These concepts have evolved into many applications, including the computation of constrained configuration spaces Porta et al (2014), and we briefly describe it next oriented to our use-case.

Let our implicitly defined manifold —or surface— be defined by the equality constraint f as

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

Then, for any arbitrary *regular* point $\mathbf{x}_i \in \mathcal{X}$, we can find an orthonormal tangent basis, $\boldsymbol{\Phi}_i$, that satisfy

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

Following the notation by Porta et al (2014), we are in the special case where $n = 3$ and $k = 2$, hence I is the 2×2 identity matrix, and $\boldsymbol{\Phi}_i$ is a 3×2 matrix. This basis serves as an approximation of the commonly known exponential, $\mathbf{x}_j = \phi_i(\mathbf{u}_i)$, and logarithmic $\mathbf{u}_i = \phi_i^-1(\mathbf{x}_j)$, mappings, that satisfy the condition $\mathbf{x}_i = \phi_i(\mathbf{0})$, where $\mathbf{u}$ are the coordinates of a local parametrization of the manifold around the point $\mathbf{x}_i \in \mathcal{X}$. This bijective map is formally known as a chart $\mathcal{C}_i$ —the former *disks*. Then, in practice, a point in the parameter space, $\mathbf{u}_i$, is exponentially mapped using a two-step procedure. First the point is transformed to the ambient space as

$$\mathbf{x}_j^i = \mathbf{x}_i + \boldsymbol{\Phi}_i\mathbf{u}_i, \tag{12}$$

where the upper index is for tracking the current basis. And, second solving the orthogonal projection defined by the system

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

for instance, using a Newton procedure initialized at $\mathbf{x}_j = \mathbf{x}_j^i$, if no better guess is available. In our case, we project using a gradient descent like method though. The validity region of a chart, in this context, is related to how safe is to compute this mapping. Typical bounds are related to the local curvature and distance from the tangent space to the manifold. However, these features are not precisely known in our scenario, so this is one deviation from the exact chart creation, that will be described in the next section.

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 whether they have been intersected from all directions or not, respectively. For instance, the initial chart is by definition unbounded. This coordination yields the concept of an atlas $\mathcal{A}$, a collection of properly coordinated charts, and it completely covers the manifold when there are no unbounded charts. Since we already differ on the validity region of each chart, we naturally also differ on their coordination.

Due to our assumptions at the beginning of the section, the manifold $\mathcal{X}$ is smooth everywhere, and we should not expect singularities either. The function, of course, exists for any point in the ambient space as $f : \mathbb{R}^3 \to \mathbb{R}$, as well as the tangent basis approximation of the exponential and logarithmic mapping, and related concepts.

Up to this point, we are able to compute an atlas $\mathcal{A}$ of an implicitly defined manifold —again, or surface—, $\mathcal{X}$, given at least a point $\mathbf{x}_i \in \mathcal{X}$ (or very close to it so we can project it using the tangent basis approximation). However, nothing has been said about what would be the preferred direction from the initial chart $\mathcal{C}_i$ to expand. If we are computing the atlas exhaustively, we can choose randomly, since we are expecting to cover it all anyway. However, if one wishes to go from point $\mathbf{x}_i \in \mathcal{X}$ to point $\mathbf{x}_j \in \mathcal{X}$ always being in the manifold, perhaps there is no interest in computing the full atlas, but only the interesting part of the path that connect them. Jaillet and Porta (2013) successfully combines the RRT technique upon this idea to compute collision-free paths of constrained mechanisms. The main difference with our strategy is we configure the RRT for exploration of uncertain regions, that is, the atlas tends to go towards regions of the predicted surface that need to be improved via a tactile exploration, which it is actually the goal RRTs were proposed for at first.

3.3 Equipment specification and limitations

Tactile exploration requires a probe able to measure contact points on the surface of two rigid bodies, and desirably the normal at that contact point too. There are mainly two low-level sensor types to this end: 1) tactile sensor arrays and 2) intrinsic tactile sensors[1]. 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 sensors has been widely used in the literature due to its multi-contact capability. However, in an scenario where a surface is not known precisely, it would require that the array be on top of a deformable body to exploit this feature at most. But the deformation then becomes hard to measure, yielding useless point measures. Consequently, they are typically mounted on top of quasi-rigid surface, also for safety of the cells, but in our scenario, we would be wasting the multi-contact capability. The second type is composed of a 6-axis force-torque sensor and a known shape on top of it, typically one that allows the computation of the contact point and force in closed-form (Bicchi et al 1993). These are single-contact sensor with the resolution being that of the force-torque sensor measurement, which implicitly is the resolution of the A/D conversion of the internal load cells, in general smaller than the resolution of a tactile sensor array.

A consideration on both types is that, for tactile exploration, they need to be mounted in a robot with at least 6 degrees of freedom, to allow the exploration to happen in different orientations w.r.t. the explored object. The mobility can be further increased if the object to be explored is also grasped by a movable robot with some degree of freedom. Inherently requires that the gripping device be soft and adaptive, but firm when grasping, since the initial assumption is that the shape is not known.

In both sensor types, the reachability spaces is of course limited to the size of the probe. It is worth considering, however, that with the intrinsic tactile sensor type, one can build a very small tip to reach tiny spaces.

[1] There are works reporting the use of a proprioceptive system, but we stick with this two given its wider use.

3.4 Problem definition

Given all these ingredients, particularly that of having a probabilistic way to represent the surface of an object with a GP over initial and partial observations, the problem at hand we present is: can we devise an exploration strategy *intrinsic* to the model that exploit its probabilistic nature to suggest points[2] for a tactile exploration such that the predicted shape from the model improves? The following section provides details of our proposed solution, the GPAtlasRRT strategy.

4 The GPAtlasRRT strategy

Form the preceding section, probably the reader has envisioned already the dots to connect from both ingredients, the Gaussian process implicit surfaces and the AtlasRRT methodology. The next paragraphs formally describes these connections as the GPAtlasRRT strategy, to give one possible answer to the question raised in the problem statement: an strategy that suggest the best-next tactile exploration candidates to improve the object surface representation, exploiting its probabilistic nature.

The method starts from an initial and partial observations of the object surface. This is generally provided by vision due to its ability to cope with several points at once. To this end, we assume there is a way to segment the object points out of the scene, that is, points on the object surface[3]. However, there could be no information at all, and the probe naively go towards the palm of the gripper holding the object to get a first initial observation, and start from there. As one would expect, this increases the time to get an accurate shape. This initial observations go directly to form the set $\mathcal{S}^0$ to have the object surface model, $\mathcal{GP}$, model provided by (2). Since these points are axiomatically on the surface with some degree of uncertainty due the sensor noise, we can compute charts on each of them. Algorithm 1 describes the next-best candidate points for a tactile exploration that to improves input model, $\mathcal{GP}$, up to a predefined variance, $\mathbb{V}_{\max}$. The first thing to do is to SELECTSURFACEPOINT (line 1) randomly to get a point $\mathbf{x}_i \in \mathcal{X} \subset \mathcal{S}^{0}$[4]. This is the starting point where the first chart is created, invoking the CREATECHART function (line 2). The generated data structure contains the center, $\mathbf{x}_i$, the orthonormal tangent basis provided by (11), $\boldsymbol{\Phi}_i$, note that $\nabla f(\mathbf{x})$ is equivalent to (8), its size, ρ_i, and a set of points in the tangent space, $\mathcal{U}_i$. Two things are noticeable here that differentiate us from the original AtlasRRT algorithm. Firstly, the size of a chart, which is a similar to the validity region notion, is inversely proportional to the variance at the chart center, namely,

Algorithm 1: The GPAtlasRRT strategy

```
   $\mathcal{P} \leftarrow$ GPAtlasRRT($\mathcal{M}$, $\mathbb{V}_{\max}$)
   input  : A Gaussian Process model, $\mathcal{M}$ and the set of
            parameters $\Omega$, 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      $\mathcal{C}_j \leftarrow$SELECTCHART($\mathcal{A}$)
6      $\mathbf{x}_k \leftarrow$EXPANDCHART($\mathcal{C}_j$, $\mathcal{GP}$)
7      $\mathcal{C}_k \leftarrow$ CREATECHART($\mathbf{x}_k$, $\mathcal{GP}$)
8      $\mathcal{A} \leftarrow$ADDCHART($\mathcal{A}$, $\mathcal{C}_k$)
9      if $\mathbb{V}[f(\mathbf{x}_k)] > \mathbb{V}_{\max}$ then
10         $\mathcal{P} \leftarrow$ GETPATH($\mathcal{C}_i$, $\mathcal{C}_k$)
11         return $\mathcal{P}$
12 return $\varnothing$
```

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

This size is actually the radius of a ball centered at $\mathbf{x}_i$, whose intersection with the tangent space yields *disk*. The motivation behind this choice is that the more certain a point is to be on the surface, the larger the covered region of its chart on the predicted shape, whereas the more uncertain is associated to the center, we prefer to perform smaller exploratory steps. And secondly, a number proportional to the size of points are sampled uniformly and randomly in the annulus region of the disk with internal and external radius being 0.8ρ and ρ, respectively. The cardinality of this point-set in the tangent space is proportionally to its size, namely,

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

This leads to the fact that the larger the chart, the higher the resolution of directions to expand, hence more samples are needed to have a good quantization of it. Another advantage of working with a normalized and offset-free set, as mentioned in Subsection 3.1, is that the parameters that makes the latter two expression to be equalities are tuned once, and remain fixed.

The algorithm continues with the addition of this first chart to the atlas, $\mathcal{A}$, that becomes the root node of an exploration tree (line 3). The question whether

[2] Note that the points can be isolated (poking) or arranged in a path (sliding), indistinctly.

[3] We provide technical details of our implementation in a real scenario in Subsection **??**

[4] In theory, we can start from all points in the surface, but in practice, this requires too many multi-processes to be properly synchronized

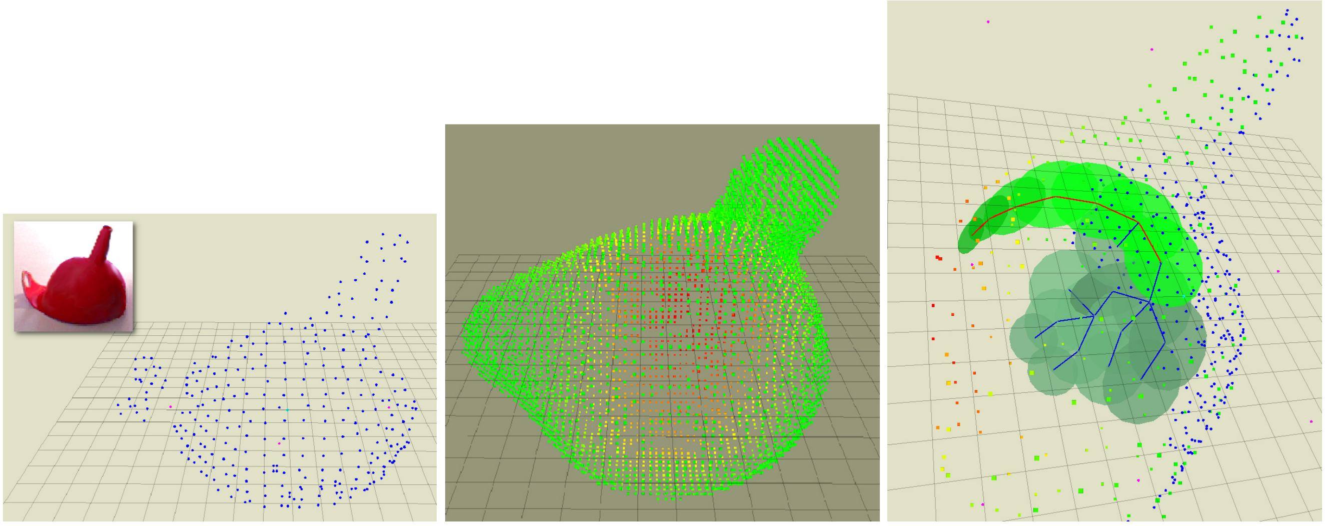

Fig. 2 A funnel (left-upper corner) is first seen by a depth camera. The segmented 3D points are sown in blue in the left figure to form the training set $\mathcal{S}^0$. The predicted shape by the GP on this set is shown in the middle obtained via the marching cube algorithm. However, the GPAtlasRRT 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.

an atlas isExpandable or not (line 3) is readily answered by checking whether there is at least one chart with $\#\mathcal{U}_j \neq 0$ or not. The `false` condition implies that we have covered completely the predicted surface, exiting with no candidate for the best-next tactile exploration (line 12). The `true` condition then makes the flow to continue within the **while** loop for the iterative exploration. The first step is to selectChart (line 5) to expand from the atlas. This action takes place randomly among all current charts whose point-set $\mathcal{U} \neq \emptyset$ using a trade-off proportion between depth-first (or selecting the previously expanded chart with probability $p = 0.4$) and breadth-first (or selecting any other chart with probability $1 - p$) for the tree expansion. Note that, in the first iteration, $\mathcal{C}_i$ is the only one, and it is surely expandable. Next, the expandChart operation on the selected chart (line 6) chooses among all points $\mathbf{u}_{j,t} \in \mathcal{U}_i$, the one with the largest variance in ambient space, that is,

$$\mathbf{u}_j^* = \arg\max_{\mathbf{u}_j \in \mathcal{U}} \mathbb{V}[f(\phi_j(\mathbf{u}_j))], \tag{16}$$

where the tangent basis approximation of the mapping $\phi_j(\cdot)$ is used as given in (12). The point $\mathbf{x}_k^j = \phi(\mathbf{u}_j^*)$ is then projected to the predicted surface solving (13), as mentioned, using a gradient-descent-like method to obtain the point $\mathbf{x}_k \in \mathbf{X}$. A chart, $\mathcal{C}_k$, is then created from this point and added to the current atlas (lines 7–8). When a new chart is added to the atlas, the corresponding set $\mathcal{U}_j$ of all charts are reduced by discarding the points that fall within the validity region of the charts existing in the atlas according to our definition, that is, if they fall within the ball of radius ρ_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. Finally, the expected variance of the center of the last chart added is compared against the input parameter $\mathbb{V}_{\max}$ (line 8). The `false` condition of the **if** statement makes the algorithm to continue with the atlas expansion, whereas the `true` condition return the next-best exploration path, $\mathcal{P}$ (lines 10–11). Recall that this path is in the predicted shape of the object, therefore the control to follow it must be compliant to avoid damages. The new tactile observations tells whether the predicted shape is good or not by increasing the training set, $\mathcal{S}^0$, to automatically reduce the variance along the path, and improve the overall uncertainty of surface model. The next section traces how the GPAtlasRRT strategy can be embedded in a tactile exploration scenario.

4.1 Tactile exploration for object surface modeling via the GPAtlasRRT strategy

Algorithm 2 presents the pipeline model of a tactile exploration to model the surface of an object driven by the GPAtlasRRT strategy. As anticipated in the previous section, the initial and partial observation can be taken from visual input or a naive probe (lines to compute the model 2–5). From this point on, we plan the next-best tactile exploration, $\mathcal{P}$, (line 7), and pass this information to the exploratory probe. Please, note that this exploration can done by a real robot or simulated given a ground truth, a notion that is used for the method validation in Section 5.

Thus, the exploratory probe system APPROACHTO the exploration path, typically to the first point with low uncertainty when *sliding* or to the final point with high uncertainty when *poking* with the goal being a safety distance away from the predicted surface in the direction normal at the corresponding point. This phase can be done with a position control scheme and collision avoidance motion planning. The latter is an interesting topic, since the object shape is not known. But it can be predicted with the current state of the model to some degree of uncertainty. A coarse point cloud can be computed out of the model and build a probabilistic collision map based on the expected variance of the function values at the point cloud. Next, we proceed to PROBEOBJECT following the next-best path, $\mathcal{P}$, to get observations of points being on or outside of the surface, and labeled accordingly altogether in the exploration log set, $\mathcal{S}^{0+}$ (line 10). In this phase, the collision avoidance motion is disable and the probe moves compliantly, like in a contour-following set-up, and uses the tactile sensing to detect whether the probe is contacting or not with the object surface. Once the end of the path is reached, the training set is accordingly updated keeping the old training set, and applying the normalization and offset-free operations after the addition of new data (line 11). This is then used to recompute a better model of the object surface, $\mathcal{GP}$ (line 12). Simultaneously (using the old model) or afterwards (using the updated model), we engage again the position control and collision avoidance motion planning to take the exploratory probe to its rest position (line 13). When the GPATLASRRT returns an empty path, $\mathcal{P} = \varnothing$, then we say that the predicted shape by the returned model, $\mathcal{GP}$, has at most a variance of $\mathbb{V}_{\max}$ for any point on the surface, that is, $\mathbb{V}[f(\mathbf{x})] < \mathbb{V}_{\max}$, $\forall \mathbf{x} \in \mathcal{X}$. The next section presents two implementations of this algorithm, one that we use for validation, where the PROBEOBJECT is performed by ray-casting on object meshes (ground truth), and another for testing, where our Vito robot is equipped with a an exploratory probe to execute this action.

5 Validation and experiments

5.1 Apparatus

Ist this described in sec 3.1 ?

We repeat the exploratory probe given in Rosales et al (2014). The probe is composed of a semispherical tip (radius 2cm) on top of an ATI Nano 17 and an in-parallel passive compliant coupler to safely attach it as the end-effector of the 7 degrees of freedom KUKA LWR 4+ robot arm.

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 Gaussian process, $\mathcal{GP}$.
 1 if isEmpty($\mathcal{Z}$ then
 2     $\mathcal{S}^0 \leftarrow$ naiveProbe()
 3 else
 4     $\mathcal{S}^0 \leftarrow$ segmentObject($\mathcal{Z}$)
 5 $\mathcal{S} \leftarrow$ generateTrainSet($\mathcal{S}^0$)
   $\mathcal{GP} \leftarrow$computeModel($\mathcal{S}$)
 6 while true do
 7     $\mathcal{P} \leftarrow$GPAtlasRRT($\mathcal{GP}, \mathbb{V}_{\max}$)
 8     if $\mathcal{P} \neq \varnothing$ then
 9         ApproachTo($\mathcal{P}$, $\mathcal{GP}$)
10         $\mathcal{S}^{0+} \leftarrow$probeObject($\mathcal{P}$)
11         $\mathcal{S} \leftarrow$ updateTrainSet($\mathcal{S}$, $\mathcal{S}^{0+}$)
12         $\mathcal{GP} \leftarrow$computeModel($\mathcal{S}$)
13         MoveAway($\mathcal{GP}$)
14     else
15         return $\mathcal{GP}$
```

The object is grasped by a soft-hand. The assumption that the object is unknown holds to the adaptability of the hand. There is no need to have a precise model of the object, but a rough approximation of the shape. The hand softness will do the rest. This setup complies with the specifications given in Sect. **??**.

5.2 Results

To write

6 Conclusions and future work

To write

We have developed...

As an immediate future research is the study of manifold Gaussian Process as presented by Calandra et al (2014), so the application of the AtlasRRT as presented by Jaillet and Porta (2013) can be applied in a more precise manner.

Acknowledgements The authors would like to thank E. Farnioli for the fruitful discussions and initial matlab implementations about Gaussian Processes, as well as to G. Santaera for the support with the sensorized glove.

References

Allen P, Bajcsy R (1987) Robotic object recognition using vision and touch. In: Proceedings of the 9th International Joint Conference on Artificial Intelligence, pp 1131–1137

Allen PK, Michelman P (1990) Acquisition and interpretation of 3-D sensor data from touch. IEEE Transactions on Robotics and Automation 6(4):397–404

Bajcsy R (1988) Active perception. Proceedings of the IEEE 76(8):966–1005

Bajcsy R, Lederman S, Klatzky RL (1989) Machine systems for exploration and manipulation a conceptual framework and method of evaluation. Tech. Rep. MS-CIS-89-03

Bicchi A, Salisbury JK, Brock DL (1993) Contact sensing from force measurements. The International Journal of Robotics Research 12(3):249–262

Bierbaum A, Rambow M, Asfour T, Dillmann R (2008) A potential field approach to dexterous tactile exploration of unknown objects. In: 8th IEEE-RAS International Conference on Humanoid Robots, pp 360–366

Bjorkman M, Bekiroglu Y, Hogman V, Kragic D (2013) Enhancing visual perception of shape through tactile glances. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp 3180–3186

Calandra R, Peters J, Rasmussen CE, Deisenroth MP (2014) Manifold gaussian processes for regression. URL http://arxiv.org/abs/1402.5876

Caselli S, Magnanini C, Zanichelli F, Caraffi E (1996) Efficient exploration and recognition of convex objects based on haptic perception. In: IEEE International Conference on Robotics and Automation, pp 3508–3513

Dragiev S, Toussaint M, Gienger M (2011) Gaussian process implicit surfaces for shape estimation and grasping. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp 2845–2850

Dragiev S, Toussaint M, Gienger M (2013) Uncertainty aware grasping and tactile exploration. In: IEEE International Conference on Robotics and Automation, pp 113–119

Faugeras O, Hebert M (1983) A 3-d recognition and positioning algorithm using geometrical matching between primitive surfaces. In: Proc. 8th Int. Joint Conf. Artif. Intell., pp 996–1002

Grimson WEL, Lozano-Perez T (1984) Model-based recognition and localization from sparse range or tactile data. J Robot Res 3(3):3–35

Henderson ME (1993) Computing implicitly defined surfaces: Two parameter continuation. Tech. Rep. 18777, T. J. Watson Research Center, IBM Research Division

Jaillet L, Porta JM (2013) Path planning under kinematic constraints by rapidly exploring manifolds. IEEE Transactions on Robotics 29(1):105–117

Kragic D, Christensen HI (2002) Survey on visual servoing for manipulation. Tech. Rep. ISRN KTH/NA/P-02/01-SE, Stockholm, Sweden

LaValle SM (2011) Motion planning. IEEE Robotics & Automation Magazine 18(1):79–89

Li M, Hang K, Kragic D, Billard A (2016) Dexterous grasping under shape uncertainty. Robotics and Autonomous Systems 75

Meier M, Schöpfer M, Haschke R, Ritter H (2011) A probabilistic approach to tactile shape reconstruction. IEEE Transactions on Robotics 27(3):630–635

Moll M, Erdmann MA (2003) Reconstructing the Shape and Motion of Unknown Objects with Active Tactile Sensors, chap 17, pp 293–310. Springer Tracts in Advanced Robotics

Petrovskaya A, Khatib O (2011) Global localization of objects via touch. IEEE Transactions on Robotics 27(3):569–585

Porta J, Ros L, Bohigas O, Manubens M, Rosales C, Jaillet L (2014) The CUIK suite: Analyzing the motion Closed-Chain multibody systems. IEEE Robotics &amp; Automation Magazine 21(3):105–114

Rasmussen CE, Williams CKI (2006) Gaussian Processes for Machine Learning. The MIT Press

Roberts K (1990) Robot active touch exploration: constraints and strategies. In: IEEE International Conference on Robotics and Automation, pp 980–985

Rosales C, Ajoudani A, Gabiccini M, Bicchi A (2014) Active gathering of frictional properties from objects. In: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp 3982–3987

Shekhar S, Khatib O, Shimojo M (1986) Sensor fusion and object localization. In: IEEE International Conference on Robotics and Automation, pp 1623–1628

Sommer N, Li M, Billard A (2014) Bimanual compliant tactile exploration for grasping unknown objects. In: IEEE International Conference on Robotics and Automation, IEEE, pp 6400–6407

Williams O, Fitzgibbon A (2007) Gaussian process implicit surfaces. In: PASCAL - Pattern Analysis, Statistical Modelling and Computational Learning - Gaussian Processes in Practice Workshop