

GPAAtlasRRT: A probabilistic next-best tactile action strategy for object shape modelling and exploration

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 grasps 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 paper, we firstly represent the shape of an object as a Gaussian Process (GP). Then, using the concept that the 0-levelset of the GP — the surface of the approximated object — is an implicitly defined manifold, we extend the AtlasRRT algorithm to simultaneously: (i) build an atlas that locally parameterizes the object via continuation, and (ii) use the atlas to devise directions for its expansion that maximize information gain suggesting the next best touch. We integrate this strategy in a framework for iterative learning the object shape and implement a tactile exploration scenario in a bimanual setting. The experimental results confirm that the devised methodology yields better models and does it with a reduced number of touches.

Keywords Active exploration · Next-best tactile action · Shape modelling

1 Introduction

One of the main reasons that makes autonomous grasping tasks very challenging is that object properties required for grasp planning like shape and surface properties are not known *a priori*. This requires robots to perceive objects around them based on sensory information.

8 < - - - - 8 < - - - - 8 < - - - - 8 < - - - -

General guidelines : ?

Active visual perception: ?

Vision-based exploration is the most studied, perhaps due to its non-invasive nature, which avoids the contact between rigid bodies which is the cause of most headaches in physics modelling, simulation and control.

As very well said by ?, even when initial works date back to the 80's, tactile perception has not been addressed as deeply as the non-invasive counterpart, visual perception. Besides the need of being actively controlled, tactile sensors typically required ad-hoc mechanical devices.

“Touch-based perception has not been studied in as much depth as vision because standardized touch sensors are not as easily available. In many situations, tactile sensors have to be hand crafted specifically for the robot and the task. This complicates comparisons between methods and slows progress in tactile perception. However, recently there has been a surge of interest in the field due to the necessity of touch-based perception in service applications”

Whereas ? is more interested in the object pose estimation problem, here we are more interested in the

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.
Tel.: +39 (050) 2217050
E-mail: carlos.rosales@for.unipi.it

C. Zito and J. L. Wyatt
Intelligent Robotics Laboratory, Univ. of Birmingham, UK.
Tel.: +44 (0) 121 415 8722
E-mail: cxz004@cs.bham.ac.uk

object shape modelling, sort of in the mapping of rather than localization in a SLAM problem.

On active touch sensing ?

Differentiate from active touch for localization, here another example: ?.

Justify the use of a intrinsic tactile sensor over a tactile array.

ITS are more precise and less noisy. It provides the contact normal directly in sensor frame.

Tactile arrays provide multiple-point measurements. They do not provide directly the normal in sensor frame, forward kinematics is required over noisy joint measurements, or in fixed configurations that limit exploration mobility. Up to a point that a typically a complex framework is needed to exploit its grasping and touching properties

With ITS, the computation will be trivial. The disadvantage is that it is a single-point measurement. Poking strategies, like in ?, or trajectories strategies like in ?.

On how to do classification...

Shape representation and descriptor review ?

In this work, we provide a systematic methodology to plan the next-best tactile exploratory action. The tactile exploratory action is intrinsically a contact hypothesis that need to be accepted or discarded after execution. Should the result be any of the two, it helps to improve the object shape prediction up to a pre-specified variability. In contrast to ?, we propose the same shape representation as descriptor for classification purposes using shape matching techniques (?). The scope and problem statement is detailed in Sect. 3. The proposed solution is broaden in Sect. 4. The experimental results and discusion is presented in Sect. 5. Finally, the conclusions and points deserving further attention as given in Sect. 6.

2 Related work

2.1 State of the art

The early work by ? presents a hierarchical representation of the object. The tactile exploration strategy to refine is driven by local geometry features. This is engaged by using surface tracing algorithms. In that work, it is literally said: "Given a starting and ending point on a surface, the sensor traces along the surface reporting its contact positions and normals as it moves along." However, no indication is given in how to determine those starting and ending points on the surface. ? blah blah

? follows the covariance function derived by ?. This is the closest work to ours among the references. The

main difference being the space in which the best-next exploratory action is computed and the terminal condition for the overall algorithm, and the descriptor. The Zernike moments imply an extra computation time, and lack of a probabilistic interpretation, which it is one of the good things about using Gaussian Process in first place. The exploratory actions are searched in a discrete space in the vertical direction and the approach angle, which seem extrinsic to the shape model. Using this space does not guarantee that the new observations will be on the desired shape region to be explored. Finally, the number of actions, or touches in this case, are limited to a certain number, and then ordered according to the closes 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.

These two works, ? and ? are very very similar to our approach.

More similar works, ?, ?, ?

2.2 Background

Regarding the shape representation...

? probabilistic representations of shapes

Using Gaussian Process for shape modelling is having an intensive development in recent years (????), due to its versatility to accomadate noisy information, provide smooth regression of data, and a natural way to fuse different sources of information (?).

Manifold gaussian process: ? "The quality of a Gaussian Process model strongly depends on an appropriate covariance function."

The tree is constructed incrementally from samples drawn randomly from the search space and is inherently biased to grow towards large unsearched areas of the problem Space-filling trees are Rapidly-exploring random trees ?

Continuation method for implicitly defined surfaces ?

More general continuation method used in a more complex scenario combined with rapidly-exploring random trees (?) by ?, where the idea is to explore the part of the manifold defined by several kinematic loop constraints that solves the motion planning query.

? succesfully recover non-rigid shape, means that gaussian processes can be used

? uses a squared-exponential-like covariance function. This is selection seems to be in agreement with the use is given to the shape estimation as a gradient field that drives the reach-to-grasp controller in a smooth way.

? exploits the Gaussian Process modelling the object shape to find geodesic trajectories on the surface that are later followed by an exploratory probe to gather frictional properties. However, no further use of that property is given.

? proposes similar ideas as those given in ?, in fact they follow the same covariance function for the shape modelling, but adds a local optimization step to have a more elaborated grasp controller that drives the hand to the most-likely succesful object grasp using a metric that involves the well-known Ferrari-Canny measure.

? derives the thin-plate covariance function for 3D shapes represented by implicit functions. The property of the thin-plate function to keep the tendency outside the training data is ideal for tactile exploration (?, Fig. 2)

Still not sure whether it should go in intro, soa, or background. ?

3 Scope

We rely on previous works that have proved useful object shape representations via Gaussian Process in the derivation of grasp controllers.

3.1 Equipment specification

The vision system should be able to provide an initial guess on the object location and observation points on the surface. This is not an strict requirement, since one might be completely blind and still recognize objects around (?, e.g.). However, the use of an initial set of observation can be done quickly using RGBD sensors to speed up the overall process.

It is true that mounting the camera as the end-effector of a robot might allow a full object scanning, but there might be cases where this might not be possible either due to reaching limitations or even sensing capabilities on reduced spaces.

3.2 Assumptions

We assume that a point cloud of a segmented object is provided. However, we provide an optional pre-processing step that shows a reasonably way to do it when the object model is unknown.

Workspace. This is not to be thought as the workspace of a robot, but of the strategy algorithm that works on top of the object shape model. That is, we assume that we are modelling and exploring household objects that can be grasped and manipulated using a human-sized

hand. This is specially useful in cases where the predicted shape is not bounded using the given observations, so this workspace will shrink the predicted shape to prevent the robot from going to an empty space, or worst, hitting undesirebly.

We consider contact to appear when the force torque sensor measures a value higher than 1N. The robot motion is set slow such that inertial forces are not reflected on the measurements. This is done for safety reasons, since in the phase when the robot is approaching, there is a stop signal if this threshold is superated.

3.3 Problem statement

Considering the equipment limitations 3.1 and the assumptions 3.2, the problem for which we propose a solution can be stated as:

Given a point cloud of an object, \mathcal{O} , find a suitable representation for the shape that can be exploited for tasks such as object identification and grasping, and a coherent strategy to improve the representation independent of external references, i.e. intrinsic or exploiting the representation, and flexible enough to generate different exploratory actions such as poking points and sliding paths.

3.4 Gaussian Process Regression for Implicit Surfaces

Many robotics applications suffer of lack of complexity in the model used to represent the problem at hand. A new trend in robotics is to encode states and observations in a probabilistic framework to cope with novelty in the face of uncertainty. Our work follows such an approach in the sense that we integrate, in an online fashion, multi-sensory inputs (here vision and tactile clues) in a stochastic model.

This section formulates the problem of object shape modelling while simultaneously planning information-gathering actions for unknown objects. By unknown objects, we mean that the robot does not have priory knowledge of the geometrical features of the object to be explored. This section proceeds as follows: we first introduce a compact framework for shape estimation, here a Gaussian Process (GP), and we show how to extend such framework to estimate implicit surfaces and its first-order quantities (surface normals). Secondly, we introduce how to use such a first-order information to build an affine space of the object's surface, this procedure is called in topology as Atlas, and we also present how to exploit the affine space to compute more robust (tactile) exploration strategy in the face of shape uncertainty.

Let \mathcal{S} be a training dataset of N observations, $\mathcal{S} = \{(\mathbf{x}_i, y_i) | i = 1, \dots, N\}$, where $\mathbf{x}_i \in \mathbb{R}^D$ denotes the i^{th} input vectors and $y_i \in \mathbb{R}$ the associated scalar output or target. We collect the N inputs vectors in a $D \times N$ matrix X and the output target in a $N \times 1$ vector \mathbf{y} , so that we can rewrite the training dataset in a more compact way as $\mathcal{S} = \{(X, \mathbf{y})\}$. We are interested in making inference about the mapping function $f: \mathbb{R}^D \rightarrow \mathbb{R}$ between the input vectors and the targets. Assuming a linear regression model

$$y_i = f(\mathbf{x}_i) + \epsilon$$

where $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$ is an independent identically distributed Gaussian noise with zero-mean and variance σ_n^2 .

A Gaussian Process framework for regression (GPR) is a common choice to approximate $f(\mathbf{x})$, in which the targets y_i are assumed to be drawn from a zero-mean multi-variate Gaussian distribution with a covariance matrix whose a function of the input vectors. The output distribution can be written as

$$(y_1, \dots, y_N | \mathbf{x}_1, \dots, \mathbf{x}_N) \sim \mathcal{N}(0, K(X, X) + \sigma_n^2 I)$$

The covariance function expresses somehow the notion of nearness or similarity, for which points in the input space \mathbb{R}^D that are close would likely produce similar outputs. The choice of a kernel is a crucial ingredient for a GP and a vast part of the literature has investigate this problem, sometimes referred as the *kernel trick*. Following previous work on implicit surfaces estimation we chose the *thin plate* kernel, which is defined as

$$K(\mathbf{x}_i, \mathbf{x}_j) = 2\|\mathbf{x}_i - \mathbf{x}_j\|_2^3 - 3R\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 + R^3$$

where $R = \max(\|\mathbf{x}_i - \mathbf{x}_j\|_2, \forall \mathbf{x}_i, \mathbf{x}_j \in \mathcal{S})$, is the largest pairwise distance between the input vectors of the training dataset \mathcal{S} . Given a new point \mathbf{x}_* , it is possible to query the GP to compute the estimate of $y_* = f(\mathbf{x}_*) + \epsilon$ with a measure of confidence expressed by the associated variance $\mathbb{V}[y_*]$ by computing the following equations:

$$y_* = \mathbf{k}_*^\top (K + \sigma_n^2 I)^{-1} \mathbf{y}$$

$$\mathbb{V}(y_*) = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^\top (K + \sigma_n^2 I)^{-1} \mathbf{k}_*$$

where we introduced a more compact notation where, for a single query point $\mathbf{x}_* \in \mathbb{R}^D$, $\mathbf{k}_* = K(\mathbf{x}_*, X)$ is the $N \times 1$ covariance vector between the query point and the training dataset X . Similarly, $K = K(X, X)$ is the covariance matrix evaluated on the training dataset and $k(\mathbf{x}_*, \mathbf{x}_*)$ is a scalar value denoting the variance between the query point and itself.

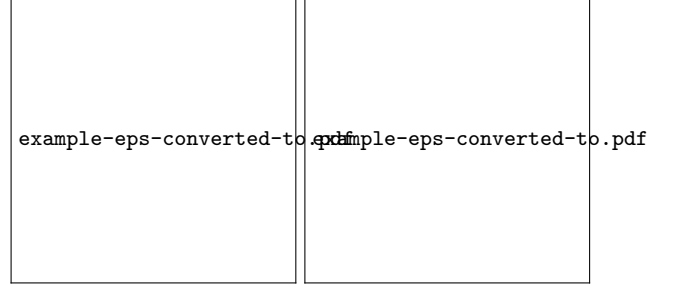


Fig. 1 A picture of the in-hand object segmentation using a soft-adaptive hand on a 7-dof arm in the real scene (left) and the resulting point cloud (right).

4 Solution

First, we provide a pre-processing step with the objective of having a segmented point set of the object. This is then used to create the probabilistic shape representation. This is then exploited in

4.1 Object Segmentation

(?, Sec. III.A) provides good candidates for object segmentation that are well oriented to object grasp, manipulation, and for our purposes, exploration. In fact, the combination of their “Table Plane Estimation” with their “Volume-Based Segmentation” yields the ubiquitous tabletop object segmentation (?). However, to improve the reachability of the exploration, we prefer to hold the object in one hand-arm system and explore with another, as described in Sec. 3. Thus, our selected approach for object segmentation is simpler than those approaches. The difficulties is in measuring the configuration of a soft and adaptable gripper that keeps the object in position to filter out the points belonging to the hand-arm system. For this purpose, one can rely in body-type measurements instead of traditional encoders, as described by ?. Once the hand-arm system pose is measured, then a trivial passthrough filter with slightly scaled bounding boxes of the robot geometry is used to obtain a point cloud of the object isolated from the scene. Algorithm 1 shows the pseudo-code of this procedure, indicating the actual. Additionally,

It is worth noting, that this module is here just to show how visual and tactile data might be merged into a single shape model. But as a matter of fact, the initial training point set can be empty and start by simply probing naively towards the gripper to initialize the training set.

The LISTENTOROBOTSTATE(\cdot) method considers the measurement of soft-adaptive mechanism

Algorithm 1: In-hand object segmentation.

segmenInHand($\mathcal{R}, \kappa, \delta$)
input : The point cloud of the scene \mathcal{P} , the robot model, \mathcal{R} , its visual geometry inflation factor, κ , and the downsampling resolution, δ .
output: The point cloud of the object isolated from the scene, \mathcal{O} .

```

1  $\mathbf{s} \leftarrow \text{LISTENTOROBOTSTATE}(\mathcal{R})$ 
2  $\mathcal{B} \leftarrow \text{COMPUTEBOUNDINGBOXES}(\mathcal{R}, \mathbf{s}, \kappa)$ 
3  $\mathcal{O} \leftarrow \text{APPLYPASSTROUGHFILTER}(\mathcal{P}, \mathcal{B})$ 
4  $\mathcal{O} \leftarrow \text{APPLYDOWNSAMPLEFILTER}(\mathcal{O}, \delta)$ 
5 return  $\mathcal{O}$ 

```

4.2 Shape modelling

What makes a good surface representation?

- Accurate (we handle this with probability)
- Concise (we might want this)
- Intuitive specification (we don't need this)
- Local support
- Affine invariant
- Arbitrary topology (we need this)
- Guaranteed continuity (we need this)
- Natural parameterization (we don't need this)
- Efficient display (we shouldn't need this)
- Efficient intersections (we could need this)

Looks like implicitly defined surfaces are the best...

How do we define implicit function?

- Algebraics
- Blobby models
- Skeletons
- Procedural
- Samples
- Variational
- Gaussian Process !
- Variational surfaces:
- Advantages:
 - Easy to test if point is on surface
 - Easy to compute intersections/unions/differences
 - Easy to handle topological changes
- Disadvantages:
 - Indirect specification of surface
 - Hard to describe sharp features
 - Hard to enumerate points on surface !
 - Slow rendering !

Except from rendering and surface explicit related things... but we don't actually need that except from debug.

4.3 Exploration strategy

We don't need all the conditions described in ?, Fig. 8.

example-eps-converted-to.pdf

Fig. 2 Different shape representations

Algorithm 2: Gaussian Process regression

createGaussianProcess(X)
input : The training data, \mathcal{X} , in the form of a point cloud.
output: The Gaussian Process that models the object shape.

```

1  $\mathcal{D} \leftarrow \text{DEMEANNORMALIZEANDLABEL}(\{\mathcal{X}, \mathbf{0}_{\text{sizeOf}(\mathcal{X})}\})$ 
2  $\text{ADDLABELEDPOINT}(\{\mathbf{0}_3, -1\}, \mathcal{D})$ 
3  $\text{ADDLABELEDPOINTS}(\{\text{SPHERE}(\mathbf{0}_3, 1.1, N), +1_N\}, \mathcal{D})$ 
4  $\mathcal{G} \leftarrow \text{DOREGRESSION}(\mathcal{D})$ 
5 return  $\mathcal{G}$ 

```

Algorithm 3: Best-next tactile action planner

exploreGPAtlasRRT($\mathcal{M}, \mathbb{V}_{max}, \Omega$)
input : A Gaussian Process model, \mathcal{M} , the variance threshold, \mathbb{V}_{max} , and the set of criteria to decide how to search and end the exploration, Ω .
output: The best next action, \mathcal{P} , in the form of a path, if any, or \emptyset otherwise.

```

1  $\mathcal{P} \leftarrow \emptyset$ 
2  $\mathcal{A} \leftarrow \text{INITGPATLAS}(\mathcal{M})$ 
3  $\mathcal{T} \leftarrow \text{INITRRTS}(\mathcal{M})$ 
4 while  $\text{ISENDCONDITIONMET}(\mathcal{A}, \Omega)$  do
5    $\mathcal{T}_i \leftarrow \text{SELECTTREETOEXPAND}(\mathcal{T}, \Omega \rightarrow \text{policy})$ 
6    $\mathcal{C}_{i,k} \leftarrow \text{SELECTCHART}(\mathcal{T}_i, \Omega)$ 
7    $\chi_{i,k} \leftarrow \text{SAMPLECHART}(\mathcal{C}_{i,k}, \Omega)$ 
8    $\mathbf{x}_{i,k} \leftarrow \text{SELECTCANDIDATE}(X_{i,k})$ 
9    $\mathbf{x}_{i,k} \leftarrow \text{GENERATE}(\mathbf{x}_{i,k}, \mathcal{M})$ 
10 return  $\emptyset$ 

```

A chart, for us, contains the center \mathbf{x}_c , such that $\mathbb{E}(\mathcal{S}(\mathbf{x}_c)) \approx 0$, the size $R = \mathbb{V}(\mathcal{S}(\mathbf{x}_c))$ and the gradient information at the center $\frac{\partial \mathcal{S}}{\partial \mathbf{x}} \approx$.

The validity are of each chart is defined as $\Phi^T \Phi \leq \cos(\alpha) \|\mathbf{u}_{j,k}\| \leq \mathbb{V}(\mathcal{S}_{j,k})$

The area covered by a given chart on the surface is never empty, and it always includes the center of the chart, \mathbf{x}_i , but its shape depends on the local shape. However, in this work, we are not interested in computing this area, since we will be traversing trees using compliance control to compensate for the errors between linear interpolation between \mathbf{x}_i and \mathbf{x}_j in 3D and actual arbitrary shape countour joining them.

A chart is selected at random with uniform distribution among the all charts in the leafs

4.4 Solution in a nutshell

The methods from the previous sections are independent from each other, being the last one, Algorithm 4, our main contribution. Here, we show an example of how these methods intercommunicate to estimate the shape of an unknown object.

Algorithm 4: Probabilistic object shape modelling

ObjectShapeExploration(P, \mathbb{V}_{des})

input : An initial point cloud of the scene, P , if any, for instance from visual object segmentation, and the desired variance, \mathbb{V}_{des} , for the overall shape estimation.

output: The estimated object shape, \mathcal{S}

```

1  $\mathcal{S} \leftarrow \emptyset$ 
2 if ISEMPTY( $P$ ) then
3    $\chi \leftarrow \text{NAIVEPROBE}()$ 
4 else
5    $\chi \leftarrow \text{SEGMENTOBJECT}(P)$ 
6  $\mathcal{S} \leftarrow \text{CREATEGAUSSIANPROCESS}(\chi)$ 
7 while true do
8    $\Gamma \leftarrow \text{EXPLOREGPATLASRRT}(\mathcal{S}, \mathbb{V}_{des})$ 
9   if  $\Gamma = \emptyset$  then
10    return  $\mathcal{S}$ 
11  else
12     $\text{APPROACHTO}(\Gamma)$ 
13     $\tilde{\chi} \leftarrow \text{PROBEOBJECT}(\Gamma)$ 
14    if WASCONTACTDETECTED() then
15       $v \leftarrow \mathbf{0}_{\text{sizeOf}(\tilde{\chi})}$ 
16    else
17       $v \leftarrow \mathbf{1}_{\text{sizeOf}(\tilde{\chi})}$ 
18     $\text{ADDLABELPOINTS}(\{\tilde{\chi}, v\}, \chi)$ 
19     $\mathcal{S} \leftarrow \text{CREATEGAUSSIANPROCESS}(\chi)$ 
20     $\text{MOVEAWAY}()$ 

```

During the APPROACHTO (line 12) (MOVEAWAY, line 20) phases, the robot uses position control and standard motion planning techniques with collision avoidance. Since we are modelling the shape, we need to ensure that everytime the robot moves close (away) from

the object, it does not collide with the object. It is tentative to use the current estimated shape, but since we are not actually computing it explicitly in our approach, we choose the bounding sphere as the collision geometry of the object. Thus the robot moves towards (away from) the surface at the contact location in the normal direction until reaching the bounding sphere. After that, a standard motion planning is used to approach the object (get to the rest position).

The PROBEOBJECT (line 13) phase is engaged once the robot is within the bounding sphere. The robot uses Cartesian impedance control, with the Cartesian force, pose and impedance set properly for the given setup. These implementation details are given in the next section. Since we don't actually know where the surface is, we need to whether the robot actually touched something or not, in order to properly label the acquisition (lines 15 and 17)

The method finishes when EXPLOREGPATLASRRT (line 8) described in Algorithm 2 has explore sufficiently the estimated shape and could not find an exploratory action Γ (line 10), i.e. the object shape is probabilistically estimated within the 95% of the confidence interval computed from \mathbb{V}_{des} .

The complete solution of the problem as stated in 3 is depicted in Algorithm 4.

4.5 Parameters and probabilistic completeness

We can safely assume that the surface has only one component when seen as a manifold.

5 Experiments

5.1 Apparatus

We repeat the exploratory probe given in ?. 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.

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. 3.1.

5.2 Results

6 Conclusions and future directions

We have developed...

As an immediate future research is the study of manifold Gaussian Process as presented by ?, so the application of the AtlasRRT as presented by ? 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.