Object Shape Estimation through Touch-based Continuum Manipulation

Huitan Mao and Jing Xiao

Abstract Object shape information is crucial for many robotic tasks. In this paper, we present an approach of estimating the shapes of unknown objects through touch-based continuum manipulation. Comparing to existing work for shape estimation that uses a conventional robot end-effector to make contact with the object, our approach offers the following advantages: 1) collecting contact points more efficiently through whole-arm wraps using a continuum manipulator; 2) explicitly taking advantage of the continuum robot proprioception to estimate the object shape both more efficiently and more accurately. Our experiments on objects with various shapes demonstrate the effectiveness of the approach.

1 Introduction

Information of object shape is crucial for many robotic tasks. A grasp can be planned to fetch an object with a known shape. When planning a path, collision can be checked against an object using its shape and configuration. Information of object shape also facilitates object detection, recognition, and pose estimation. Usually, the shape of an object is either provided as a priori knowledge or acquired by object model building through sensing, especially vision and tactile sensing.

Vision sensing has been widely used for object model building. An object appearance model can be built by either moving an RGB-D camera around the target object [1] or moving the object with a turntable [2,3]. Through robotic manipulation to change the bottom surface of a table-top object, an automatic approach interleaving perception and manipulation [4] is able to build the entire surface model of the object. However, vision sensing can be ineffective for transparent objects or in environments with poor illumination and specularity conditions.

Huitan Mao · Jing Xiao
Department of Computer Science, University of North Carolina at Charlotte, 9201 University City Blvd, Charlotte, NC 28223, e-mail: {*hmao*4,*xiao*}@uncc.edu

Tactile sensing is useful for exploring object shapes and building object models when vision is ineffective. By equiping a robotic end-effector with tactile sensors, the surface of an object can be explored with guarded moves [5] and compliant motion [6, 7]. Object recognition [8] and grasping of unknown objects [9] can also be achieved. In [10], recognition of curved surfaces through touch is achieved by matching contact points to principal-curvature-based local geometry features. Similar features are also shown to be useful in reconstructing the local surface patch by fitting a high-order polynomial [11].

In order to guide touch-based exploration, i.e., to decide where to touch next to collect contact points, there are several methods. One method uses a dynamic potential field [12], where a uniformly attractive potential field is updated as more contacts are made and generates repulsive forces to push the touch-enabled hand to visit unexplored areas. Gaussian Process (GP) [13] is used to drive active exploration into uncertain areas. In [14], discrete touch probings of the end-effector are progressively generated to reduce the uncertainty of the interest area using GP regression. In [15], it is shown that GP classification can also effectively bias the exploration towards the boundaries of the objects, which are more informative of the object shape. Extended from GP, a probabilistic model of uncertainty based on Gaussian Process Implicit Surfaces (GPIS) [16] is used to guide the active exploration and modeling of an object [5, 17] or serve as a framework of data fusion from sensors with different modalities [18].

However, using only the robot end-effector to touch a target object usually only makes a couple contact points per probing, and thus it is a slow process to collect sufficient contact points to capture the global shape of the object by changing the end-effector pose after each probe. The process also does not explicitly utilize the adjacency information of nearby contact points.

Unlike conventional robotic manipulators, continuum robots [19–24] are suitable for whole-arm manipulation to wrap around an object and for maneuvering in cluttered space. Autonomous motion planning algorithms are introduced for generating whole-arm wrapping configurations [25], force-closure wraps [26, 27], and manipulator motion constrained by the target object surface [28], for a known object. More recent work addressed object wrapping [29] and object modeling using RGBD sensing [30] by a continuum manipulator in an unknown environment.

In this paper, we address the problem of estimating the shape of an unknown object through obtaining object shape information from touch-based continuum manipulation, which has not been studied before. By wrapping around an object (see Fig. 1 for an example), a continuum manipulator makes many more contacts with an object than a conventional robot end-effector. Moreover, the manipulator shape itself in a wrapping configuration is also indicative of the object shape. Our approach, called *progressive object shape estimation through continuum manipulation* (POSE-CoM), extends the GPIS method by explicitly incorporating the continuum robot arm shape in addition to the contact points made between the robot and the object in each wrap to estimate the overall object shape. The approach is shown to be both more efficient and more accurate over existing methods for touch-based object shape estimation.

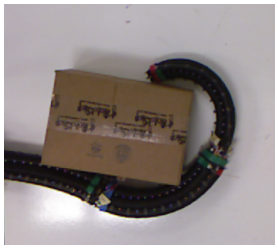


Fig. 1 A continuum manipulator wraps around a box.

The rest of the paper is organized as follows. Section 2 presents the manipulator model considered in this paper. Section 3 presents the proposed approach. Section 4 reports experimental results on objects with various shapes to demonstrate the advantages of this approach. Section 5 concludes the paper.

2 Manipulator Model

We briefly review the manipulator kinematic model used in this work. A continuum manipulator is considered as a concatenation of n circular sections when each section is not in contact or under external loading [21]. Each section $sec_i, i = 1, \ldots, n$, is described by three controllable variables: length s_i, curvature κ_i, and orientation ϕ_i, and bounded by its base point p_{i-1} and end point p_i. A local coordinate system is attached to the base point of sec_i with the z axis tangential to sec_i as shown in Fig. 2. The *arm configuration* C of an n-section continuum manipulator can be represented as $C = \{(s_1, \kappa_1, \phi_1), \ldots., (s_n, \kappa_n, \phi_n)\}$.

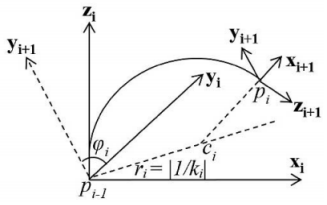


Fig. 2 Illustration of the frame of sec_i, circular arc center c_i and the control variables s_i, κ_i and ϕ_i [29].

In this work, we assume the continuum manipulator is covered with tactile sensors, which are able to detect, localize and estimate the normal and tangential direction of the contact points when the robot is in touch with the object. In simulation,

contacts between a continuum robot and an object can be efficiently detected using the collision checking algorithm in [27].

3 Progressive Object Shape Estimation through Continuum Manipulation (POSE-CoM)

An overview of our approach is presented in Fig. 3. First, **touch-based continuum wrapping** progressively moves the continuum robot to a wrapping configuration based on contact points made between the robot and the object. Next, **object shape data collection** gathers contact points and the shape information from the continuum wrap. **Shape estimation** uses the data collected so far in a probabilistic framework based on Gaussian Process Implicit Surfaces(GPIS) to estimate the overall shape of the object. **Active guidance** uses the result of estimation to decide the next continuum wrap to cover the most uncertain region of the object and collect more data for estimation. The process repeats until either the estimated object shape has low uncertainty or the robot has exhausted the possible wraps.

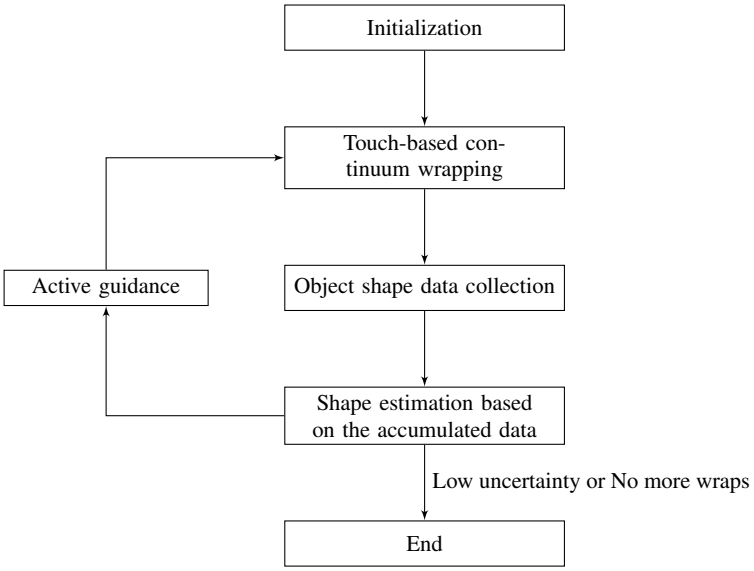


Fig. 3 Overview of the POSE-CoM approach.

3.1 Touch-based Continuum Wrapping

A continuum *wrap* of an object here is defined as a planar whole-arm grasp by the continuum manipulator around a cross section of the object, which further defines a *wrapping plane*.

Given a continuum manipulator with a fixed base, its workspace can be decomposed into discrete wrapping planes, as shown in Fig. 4. We denote the set of wrapping planes as $WP = \{wp_1, \ldots, wp_c\}$. On each wrapping plane, the continuum robot can generate a wrap of an object around the cross section on the plane. See Fig. 5 for some example wraps on different wrapping planes of an object.

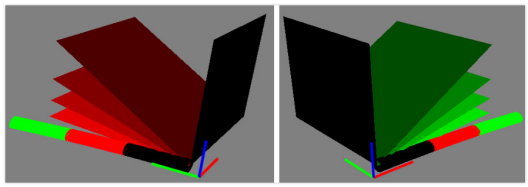


Fig. 4 The wrapping planes are systematically enumerated between the table plane (grey background) and the black planes orthogonal to the table plane. **Left:** each red plane is used to generate a clockwise continuum wrap. **Right:** each green plane is used to generate a counter-clockwise continuum wrap. Note that the initial configurations of the robot in **Left** and **Right** are different. The frame at the robot base is the global coordinate system.

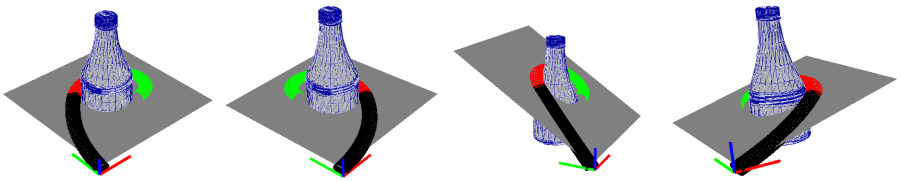


Fig. 5 A few examples of the wrapping planes (grey) and their corresponding continuum wraps. These planes also uniquely cut through the object (a bottle) and pass through the fixed robot base. The frame at the robot base is the global coordinate system.

For initialization, an initial set of wrapping planes $WP_0 \subset WP$ are randomly selected, and the continuum robot generates wraps on those wrapping planes one by one, while accumulating shape data of the object from each wrap. The data are then used to conduct shape estimation. The result is further used to guide the selection of the next wrapping plane by **Active guidance** (see Section 3.4), and so on.

Now, given a wrapping plane and an initial arm configuration, we use a motion planning strategy to generate a continuum wrap of an object progressively as guided

by the contact points made along the way [31]. The continuum manipulator alternates between the **enclosing motion step** and the **advancing motion step** on the wrapping plane until a wrap is achieved.

Enclosing motion step brings the robot into contact with the object as much as possible to create contact points. This is achieved by having the robot curve its sections one by one to make contact with the object until no further contact points can be made. Fig. 6 shows an example of such an enclosing motion.

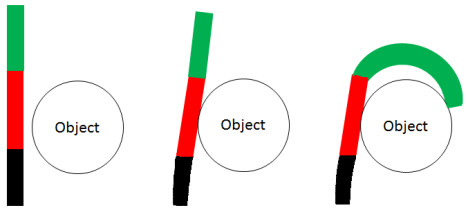


Fig. 6 Illustration of an enclosing motion. **Left**: Robot at the initial configuration. **Middle**: The robot makes first contact by curving section 1 (black). **Right**: The robot makes the second contact by curving section 3 (green) without penetrating into the object.

Advancing motion step takes advantage of the contacts made in the **enclosing motion step** to move the robot forward to a new arm configuration towards wrapping around the object. The new arm configuration is achieved by (1) moving the endpoint of each robot section a small distance from its current position along the direction $\mathbf{n}+\mathbf{t}$ of the closest contact point, where $\mathbf{n}$ and $\mathbf{t}$ denote the normal and the tangential unit vectors of the contact point respectively[1]; (2) solving the resulting arm configuration by the constrained inverse kinematics [28] corresponding to the new endpoint positions from (1).

Fig. 7 shows an example of advancing motion. The robot backbone(dashed line) is colored using black, red and green for sections 1, 2 and 3 respectively. As the result of the previous enclosing motion step, the robot is in contact with the object at the red points on sections 2 and 3. Now, for the end point of each section in contact, its new position (green) is determined by a translation of a small distance from its current position (yellow) along $\mathbf{n}+\mathbf{t}$ of the closest contact point. For section 1 that is not in contact, its new endpoint position is obtained by a small translation from the current endpoint position along the z axis of its local frame.

[1] The direction of $\mathbf{t}$ is flipped if the dot product between $\mathbf{t}$ and the z-axis of the local frame on the section endpoint closest to the contact point is negative, to ensure that the robot moves towards a wrapping configuration.

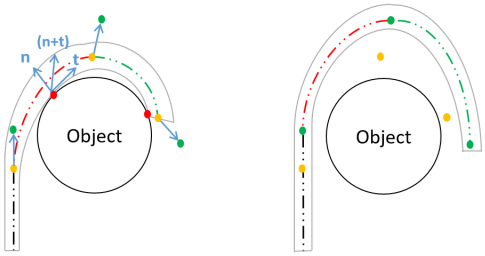
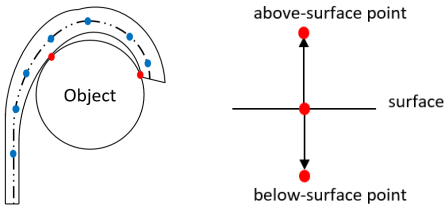


Fig. 7 Illustration of an advancing motion. **Left**: the robot section end points are indicated in yellow, the contact points are in red, and the new endpoint positions are indicated in green. **Right**: the new arm configuration solved using constrained inverse kinematics based on the new endpoint positions (green).

3.2 Object Shape Data Collection and Generation

Two types of shape data are collected in our approach (see Fig. 8(a)):

(i) contact points made between the robot and the object during wraps, and
(ii) arm points sampled along the backbone of the robot when the robot is in the configuration of a complete wrap, i.e., points indicative of the robot shape.



(a) (b)

Fig. 8 **Left**: the dashed line is the backbone of the continuum robot. The blue dots are the arm points systematically sampled on the arm backbone. The red dots are the contact points between the robot and the object. **Right**: the interpolated above-surface and below-surface points from the contact point along the contact normal.

We denote the collected points from all wraps made (both the contact points and the arm points) as $X_{col} = \{\mathbf{x}_i\}$, where $\mathbf{x}_i \in \mathbb{R}^3$, $i = 1, 2, \ldots$ A potential function value $y_i \in Y \subset \mathbb{R}$ associated with $\mathbf{x}_i$ is defined as follows, based on GPIS [16].

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

The contact points that happen on the object surfaces have $y = 0$, while the arm points sampled on the backbone of the continuum robot have $y = 1$ (since the arm does not penetrate into the object).

We further generate the above-surface and below-surface points for each contact point along the contact normal (see Fig. 8(b) for an illustration) and denote the set of such generated points $X_{gen} \subset \mathbb{R}^3$. Now let T be the set of all points that are collected and generated so far: $T = \{\mathbf{x}_i, y_i\}, \mathbf{x}_i \in X_{col} \cup X_{gen}, y_i \in Y, i = 1, 2, \ldots$

Note that initially T only contains points collected and generated from the few initial wraps. As each new wrap is conducted (based on active guidance – see Section 3.4), more points are added to T to facilitate more accurate shape estimation.

3.3 GPIS-based Shape Estimation

Estimation of an object's shape is done by finding the points with zero y value (i.e., the isosurface) in a 3D region of interest. GPIS is used to learn such a mapping $f(\mathbf{x})$: $\mathbf{x} \in \mathbb{R}^3$ to $y \in \mathbb{R}$ based on the data in T. It is fully defined by a Gaussian Process (GP) [13] with a mean function $\mu(\mathbf{x})$ and a covariance function $k(\mathbf{x}_i, \mathbf{x}_j)$, where $j = 1, 2, ...$ The prior $\mu(\mathbf{x})$ is zero. $k(\mathbf{x_i}, \mathbf{x_j})$ is chosen to be the commonly used squared exponential kernel, and a noise $\varepsilon \sim N(0, \sigma_n^2)$ is also included:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 exp(-\frac{(\mathbf{x}_i - \mathbf{x}_j)^2}{2l^2}) + \sigma_n^2 \delta_{ij} \quad (2)$$

where δ_{ij} is the Kronecker delta, which is 1 iff $i = j$ and 0 otherwise.

The hyper parameters $\{\sigma_f, l, \sigma_n\}$ are then optimized by maximizing the log-marginal likelihood [13] using the data in T, i.e., the training data.

Next, the zero-mean isosurface is extracted as the current estimation of the object shape by querying the GPIS model with τ testing points from a 3D region, which is known to contain the object or is within the reachable region of the robot.

For a testing point $\mathbf{x}_\star$, the predicted distribution is a Gaussian with the mean $\mu(\mathbf{x}_\star)$ in Eq. (3) and the variance $\sigma^2(\mathbf{x}_\star)$ in Eq. (4)

$$\mu(\mathbf{x}_\star) = \mathbf{k}_\star^T \mathbf{K}^{-1} y \quad (3)$$

$$\sigma^2(\mathbf{x}_\star) = \mathbf{k}_{\star\star} - \mathbf{k}_\star^T \mathbf{K}^{-1} \mathbf{k}_\star \quad (4)$$

where $\mathbf{k}_\star$ is a covariance matrix between m training points and τ testing points $[\mathbf{k}_\star]_{i=1\ldots m, j=1\ldots\tau} = k(\mathbf{x}_i, \mathbf{x}_{\star j})$, $\mathbf{K}$ is a covariance matrix between training points $[\mathbf{K}]_{i,j=1\ldots m} = k(\mathbf{x}_i, \mathbf{x}_j)$ and $\mathbf{k}_{\star\star}$ is a covariance matrix between testing points

$[\mathbf{k}_{\star\star}]_{i,j=1\ldots\tau} = k(\mathbf{x}_{\star i}, \mathbf{x}_{\star j})$. Note that the number of training points m increases after new wraps are conducted.

Each testing point corresponding to a zero mean of Eq. (3) is a point on the estimated object shape, and the associated posterior variance of Eq. (4) defines the uncertainty of the point due to few data points nearby or large shape change.

3.4 Active Object Exploration

Our approach actively determines the next wrapping plane to conduct another touch-based continuum wrap based on a measure of uncertainty. Denote the set of the available wrapping planes WP_a as $WP_a = WP \setminus WP_0$.

For each available plane $wp_i \in WP_a$, let P_i be the set of points on wp_i that are also on the zero-mean isosurface, then the uncertainty measure u_i of wp_i is computed as the average standard deviation $\overline{\sigma}(q), \forall q \in P_i$.

Let wp_j be the wrapping plane with $u_j = \max_i(u_i)$, then wp_j is chosen as the next wrapping plane to conduct a continuum wrap, and a new iteration in the POSE-CoM process (Fig. 3) starts. The newly collected and generated data from the new wrap are added to the training set T, and the GPIS model is in turn updated. WP_a is also updated by removing wp_j (after it has been used).

The POSE-CoM process (Fig. 3) is repeated until the maximum uncertainty u_j is below a predefined threshold or $WP_a = \emptyset$.

4 Experiments

We implemented our approach in C++ and Python under ROS and tested it on a 3.4GHz CPU with 16GB RAM. Objects of various shapes are used (Fig. 9).

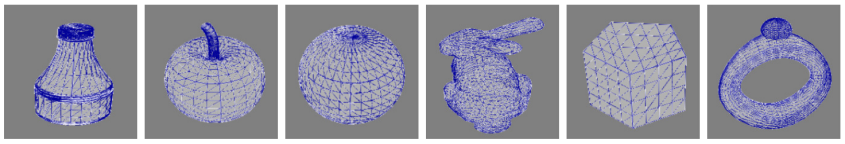
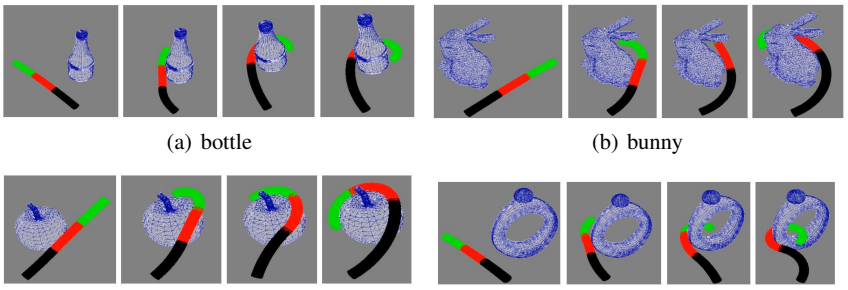


Fig. 9 Object mesh models: (left to right) bottle, apple, sphere, bunny, pentagon, vase.

To initialize the POSE-CoM process, we randomly select 4 wrapping planes in the robot workspace for the continuum robot to generate touch-based wraps and collect data. Next, a GPIS model is trained and queried in the region of $[(0.0-1.0), (0.0-1.0), (0.0-0.8)]$(dm). The simulated continuum robot has the following parameters: the width of each section is 0.5 (cm), the length of each section can vary

from 2 to 15 (cm), the curvature of each section can vary from 0 to 0.1 (cm^{-1}), and the orientation of each section can vary from $-\pi$ to π. It typically takes 3 to 15 mins to train a GPIS model. The next wrapping plane is selected according to the active strategy in Section 3.4, then touch-based continuum wrapping is conducted, and the POSE-CoM process repeats. The process is terminated when 1) the maximum uncertainty measure u_j is below a predefined threshold 0.15, or 2) all the wrapping planes in the robot workspace have been used at least once to generate continuum wraps. Fig. 10 shows the snapshots of robot motion for example continuum wraps. The accompanying video shows the animated motion of the example continuum wraps.



(a) bottle (b) bunny

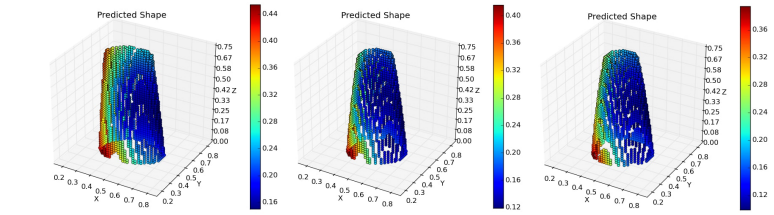
(c) apple (d) vase

Fig. 10 Snapshots of the motion of example continuum wraps. Each subfigure is a wrap from the initial configuration (left) to the wrapping configuration (right).

4.1 Shape Estimation Results

For objects with simple shape geometry, such as the bottle (Fig. 11) and the sphere (Fig. 12), with just a few wraps, the estimated shape is already similar to the actual shape. In Fig. 11, the object shape uncertainty keeps decreasing (i.e., the blue parts in Fig. 11 keep increasing) as more wraps are conducted until all available wraps are exhausted. The maximum uncertainty measure u_j starts being 0.38 and reaches 0.32 at the end. Note that the part near the origin (where the robot arm base is) is more uncertain (colored mostly red), which is due to that the robot cannot reach this area as it is too close to the robot base. In Fig. 12, note how the fifth wrap (counted from left) in Fig. 12(d) helps reduce the uncertain (red) area in Fig. 12(a). The POSE-CoM process is terminated after 7 wraps as u_j reaches 0.14 (below threshold 0.15).

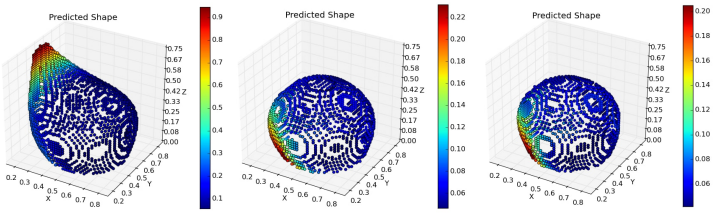
For objects with more complex shapes, such as the apple (Fig. 13), bunny (Fig. 14), pentagon (Fig. 15), and hollow vase (Fig. 18), more varied wraps are needed to better capture certain details for more accurate shape estimation results. For exam-



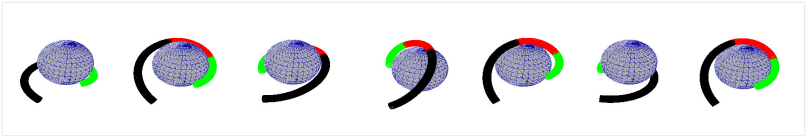
(a) 4 wraps (b) 8 wraps (c) 12 wraps

(d) Left to right: the first conducted wrap to the last conducted wrap.

Fig. 11 Shape estimation results for the **bottle**, colored by uncertainty.



(a) 4 wraps (b) 5 wraps (c) 7 wraps



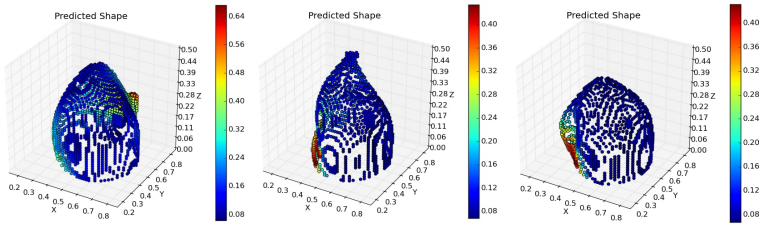
(d) Left to right: the first conducted wrap to the last conducted wrap.

Fig. 12 Shape estimation results for the **sphere**, colored by uncertainty.

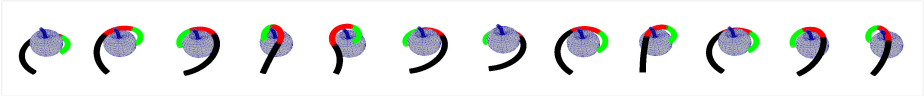
ple, wraps are needed around the stalk of the apple, the ears of the bunny, and the entrance of the hollow vase, to better capture those details. However, the GPIS modeling tends to blur the connection between a detail and the main shape because the prediction assumes smooth connection. Thus, the estimation results of a complex shape tend to resemble certain bounding envelopes of the actual shape.

Table 1 shows that a total of 500 to 900 points are used to estimate the shape of an object[2]. It would be very time-consuming if those points were collected by a conventional manipulator making point contacts with the object [5, 8, 14, 15]. However,

[2] Recall that one above-surface point and one below-surface point are generated for each contact point along the contact normal.

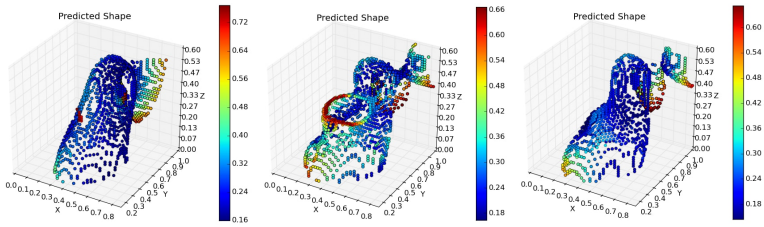


(a) 4 wraps (b) 8 wraps (c) 12 wraps

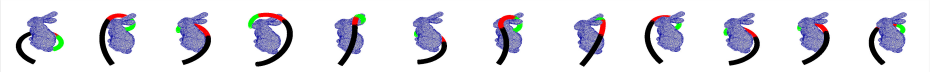


(d) Left to right: the first conducted wrap to the last conducted wrap.

Fig. 13 Shape estimation results for the **apple**, colored by uncertainty.



(a) 6 wraps (b) 9 wraps (c) 12 wraps



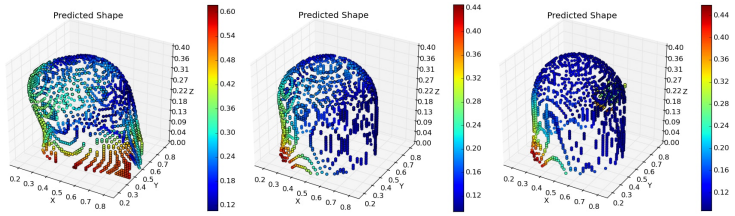
(d) Left to right: the first conducted wrap to the last conducted wrap.

Fig. 14 Shape estimation results for the **bunny**, colored by uncertainty.

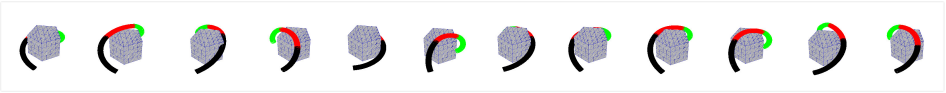
our approach is very efficient by using continuum wraps automatically generated from our motion planning algorithm. Table 1 also shows that it is very fast to plan a touch-based continuum wrap.

Fig. 16 shows the reconstructed mesh models from the shape estimation results using the Marching Cubes' algorithm [32]. As explained earlier, those models capture enveloping contours of the actual shapes because sharp details were blurred by the GPIS modeling under noisy observations.

If the reconstructed model is identical to the original mesh model of an object, given the same arm initial configuration, model location, and the wrapping plane, the final wrapping configuration C_r of the reconstructed model should be the same as the final wrapping configuration C_o of the original mesh model. Therefore, the



(a) 4 wraps (b) 8 wraps (c) 12 wraps

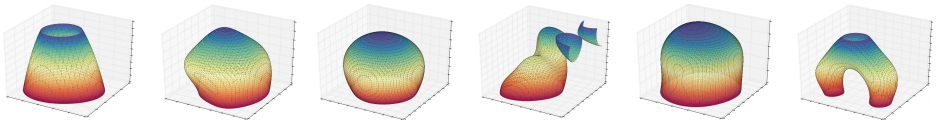


(d) Left to right: the first conducted wrap to the last conducted wrap.

Fig. 15 Shape estimation results for the **pentagon**, colored by uncertainty.

Table 1 The total number of contact points and arm points collected on each object, the total number of wraps, and the average time T_{wrap} per wrap for (motion planning + collision detection).

Object	# contact points	# arm points	# wraps	T_{wrap}(ms)
bottle	143	150	12	135
apple	259	172	12	425
sphere	267	113	7	246
bunny	264	172	12	432
pentagon	141	167	12	132
vase	153	160	12	278



(a) bottle (b) apple (c) sphere (d) bunny (e) pentagon (f) vase

Fig. 16 Object mesh models reconstructed from shape estimation results using Marching Cubes' algorithm [32], colored by elevation.

distance of normalized configurations $\|C_r{}' - C_o{}'\|$, where $C_r{}'$ and $C_o{}'$ are obtained by normalizing the component values in C_r and C_o, is indicative of how accurate the shape estimation is. We use the average distance of multiple wraps as a parameter to measure the accuracy of shape estimation indirectly, as shown in Table 2. It is clear that model quality improves as more wraps are used to reconstruct the model. Fig. 17 compares example wraps visually.

Table 2 Average distances between the wrapping configurations of a reconstructed model and the original mesh model. # wraps indicates the number of wraps used to reconstruct the model.

Object	Intermediate Model 1		Intermediate Model 2		Final Model	
	# wraps	average distance	# wraps	average distance	# wraps	average distance
bottle	4	0.31	8	0.27	12	0.27
apple	4	0.44	8	0.36	12	0.28
sphere	4	0.28	5	0.26	7	0.24
bunny	6	0.57	9	0.47	12	0.31
pentagon	4	0.34	8	0.19	12	0.16
vase	6	0.48	9	0.41	12	0.31

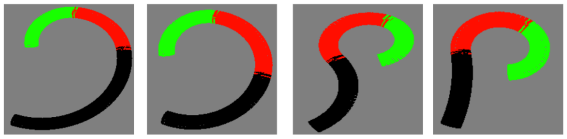


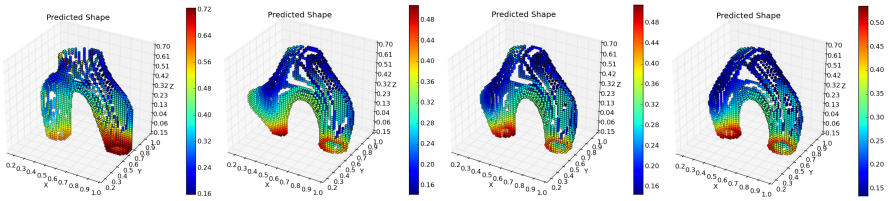
Fig. 17 Comparing example wraps on the original mesh model and the reconstructed model. (a) wrap on the original apple model (b) wrap on a reconstructed apple model (c) wrap on the original vase model (d) wrap on a reconstructed vase model.

4.2 Significance of the Arm Points

Fig. 19 shows the shape estimation results of the hollow vase using only contact points. Since the object is hollow, continuum wraps cannot make a large number of contacts, and thus the shape estimation result is inaccurate. However, the arm shape at a wrapping configuration itself is indicative of the object shape, and the arm points can be used to compensate for the lack of sufficient contact points and determine the object geometry. As shown in Fig. 18, our approach using both arm points (robot proprioception) and contact points better captured the vase shape and also captured the hollowness of the vase using as few as 6 wraps. Whereas, without using the arm points, as shown in Fig. 19, the hollowness of the vase could not be captured even with 12 wraps.

5 Conclusions

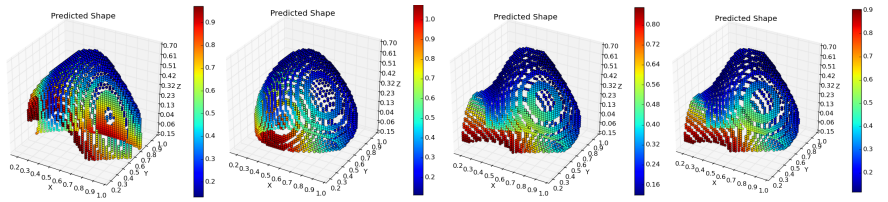
We have presented an approach POSE-CoM of progressive object shape estimation through touch-based continuum manipulation and a GPIS-based probabilistic model. Unlike shape estimation based on contact points collected through the end-effector of a conventional, articulated manipulator, shape estimation through continuum manipulation is both more efficient and effective by deliberate use of the



(a) 6 wraps (b) 9 wraps (c) 10 wraps (d) 12 wraps

(e) Left to right: the first conducted wrap to the last conducted wrap.

Fig. 18 Shape estimation results of the **vase**, colored by uncertainty.



(a) 6 wraps (b) 9 wraps (c) 10 wraps (d) 12 wraps

Fig. 19 Shape estimation results of the **vase** without using arm points, colored by uncertainty.

robot proprioception data in addition to contact information, as demonstrated in the experiments on objects of various shapes.

Acknowledgements This work is supported by the US NSF grant IIP-1439695.

References

1. R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2011.
2. C. Choi and H. I. Christensen, "3d pose estimation of daily objects using an rgb-d camera," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.
3. A. Singh, J. Sha, K. S. Narayan, T. Achim, and P. Abbeel, "Bigbird: A large-scale 3d database of object instances," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.
4. Z. Teng, H. Mao, and J. Xiao, "Automatic object modeling through integrating perception and robotic manipulation," in *International Symposium on Experimental Robotics (ISER)*, 2016.

5. S. Caccamo, Y. Bekiroglu, C. H. Ek, and D. Kragic, "Active exploration using gaussian random fields and gaussian process implicit surfaces," in *IEEE/RSJ IROS*, 2016.
6. A. M. Okamura and M. Curkosky, "Feature-guided exploration with a robotic finger," in *IEEE ICRA*, 2001.
7. Q. Li, C. Schürmann, R. Haschke, and H. J. Ritter, "A control framework for tactile servoing.," in *Robotics: Science and systems*, 2013.
8. M. M. Zhang, M. Kennedy, M. Hsieh, and K. Daniilidis, "A triangle histogram for object classification by tactile sensing," in *IEEE/RSJ IROS*, 2016.
9. N. Sommer, M. Li, and A. Billard, "Bimanual compliant tactile exploration for grasping unknown objects," in *IEEE ICRA*, 2014.
10. R. Ibrayev and Y.-B. Jia, "Recognition of curved surfaces from one-dimensional tactile data," *IEEE Transactions on Automation Science and Engineering*, vol. 9(3), 2012.
11. Y.-B. Jia and J. Tian, "Surface patch reconstruction from one-dimensional tactile data," *IEEE Transactions on Automation Science and Engineering*, vol. 7(2), 2010.
12. A. Bierbaum, M. Rambow, T. Asfour, and R. Dillmann, "A potential field approach to dexterous tactile exploration of unknown objects," in *IEEE International Conference on Humanoid Robots*, 2008.
13. C. E. Rasmussen, "Gaussian processes for machine learning," MIT Press, 2006.
14. Z. Yi, R. Calandra, F. Veiga, H. van Hoof, T. Hermans, Y. Zhang, and J. Peters, "Active tactile object exploration using gaussian processes," in *IEEE/RSJ IROS*, 2016.
15. N. Jamali, C. Ciliberto, L. Rosasco, and L. Natale, "Active perception: Building objects' models using tactile exploration," in *IEEE International Conference on Humanoid Robots (Humanoids)*, 2016.
16. O. Williams and A. Fitzgibbon, "Gaussian process implicit surfaces," *Gaussian Proc. in Practice*, 2007.
17. S. Dragiev, M. Toussaint, and M. Gienger, "Gaussian process implicit surfaces for shape estimation and grasping," in *IEEE ICRA*, 2011.
18. M. P. Gerardo-Castro, T. Peynot, and F. Ramos, "Laser-radar data fusion with gaussian process implicit surfaces," in *Field and Service Robotics*, Springer, 2015.
19. G. Robinson and J. B. C. Davies, "Continuum robots-a state of the art," in *IEEE ICRA*, 1999.
20. R. J. Webster III, J. M. Romano, and N. J. Cowan, "Mechanics of precurved-tube continuum robots," *IEEE Transactions on Robotics(TRO)*, 2009.
21. B. A. Jones and I. D. Walker, "Kinematics for multisection continuum robots," *IEEE TRO*, vol. 22(1), 2006.
22. I. D. Walker, "Continuous backbone "continuum" robot manipulators," *ISRN Robotics*, vol. 2013, 2013.
23. C. G. Frazelle, A. D. Kapadia, K. E. Fry, and I. D. Walker, "Teleoperation mappings from rigid link robots to their extensible continuum counterparts," in *IEEE ICRA*, 2016.
24. J. Burgner-Kahrs, D. C. Rucker, and H. Choset, "Continuum robots for medical applications: A survey," *IEEE TRO*, vol. 31(6), 2015.
25. J. Li and J. Xiao, "Determining grasping configurations for a spatial continuum manipulator," in *IEEE/RSJ IROS*, 2011.
26. J. Li and J. Xiao, "Progressive planning of continuum grasping in cluttered space," *IEEE TRO*, vol. 32(3), 2016.
27. J. Li and J. Xiao, "An efficient algorithm for real time collision detection involving a continuum manipulator with multiple uniform-curvature sections," *Robotica*, vol. 34(7), 2016.
28. J. Li and J. Xiao, "A general formulation and approach to constrained, continuum manipulation," *Advanced Robotics*, vol. 29(13), 2015.
29. J. Li, Z. Teng, and J. Xiao, "Can a continuum manipulator fetch an object in an unknown cluttered space?," *IEEE Robotics and Automation Letters*, 2017.
30. H. Mao, Z. Teng, and J. Xiao, "Progressive object modeling with a continuum manipulator in unknown environments," in *IEEE ICRA*, 2017.
31. H. Mao, M. M. Zhang, J. Xiao, and K. Daniilidis, "Shape-based object classification and recognition through continuum manipulation," IEEE/RSJ IROS, 2017.
32. W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," in *ACM siggraph computer graphics*, 1987.