This full text paper was peer-reviewed at the direction of IEEE Instrumentation and Measurement Society prior to the acceptance and publication.

Automatic Selection of Grasping Points for Shape Control of Non-Rigid Objects

Félix Nadon, Pierre Payeur
School of Electrical Engineering and Computer Science
University of Ottawa
Ottawa, Canada
[fnado079, ppayeur @uottawa.ca]

***Abstract*—The dexterous manipulation of non-rigid objects by robotic hands is a requirement for automating many delicate or labour-intensive tasks in various industries. This includes the ability to actively deform and shape objects to fit specifications, which is an important skill that allows, e.g., to insert a soft foam filter into a rigid enclosure. This work focuses on the in-hand shaping of non-rigid objects, providing an original model-free algorithm for automatically selecting the contact points between the fingers and the object's contour. This optimizes the initial conditions of the shaping task, allowing the desired shape to be approximated more efficiently with low degrees of freedom in the applied forces. The algorithm is validated experimentally with the Barrett hand and a variety of non-rigid objects.**

***Index Terms*—robotic manipulation, deformable objects, grasping points selection, visual servoing.**

I. Introduction

In order for robotic manipulators to reach an increased level of flexibility and gain the ability to perform varied tasks, it is necessary for them to come closer to a human-like skillset. To achieve this, an important issue that remains to be solved is the appropriate handling of non-rigid objects. This includes any objects that can change shape during or as a result of the manipulation process, such as ropes and wires, sponges, rubber, all cloth-like objects and other thin sheets, as well as a variety of organic materials including organs and living tissues. The dexterous handling of such objects will enable the use of robotic manipulators to automate many delicate or labour-intensive tasks for industrial assembly, food processing, surgery and household chores.

As shown in recent surveys [1], [2], current research involving the manipulation of non-rigid objects is moving from the handling of ropes and cloth to the often more computationally-intensive tasks of precisely shaping 3D objects and their 2D projections or contours. This work focuses on an important subproblem of this task which has not yet received much attention, namely the selection of the contact points between the robotic hand manipulator(s) and the object in order to optimize the shaping task while keeping a stable grasp and avoiding slippage.

The rest of this paper is organized as follows: section II provides an overview of the related literature while section III describes the proposed algorithm both generally and for a specific case with the Barrett robot hand. Finally, section IV presents a sample of experimental results as well as a discussion of real-time processing concerns and a highlight of some of the limitations of the algorithm, while section V provides concluding remarks and directions for future work.

II. Related Literature

A. Grasp Synthesis

Grasp synthesis algorithms can be divided in two general categories based on the relative dimensions of the manipulator and of the object to grasp, i.e., one for large objects and one for small objects (relative to the manipulator grasp size). The first category involves selecting the areas to grasp on objects which are much larger than the manipulator, without needing to consider in-hand dynamics. For instance, an automated tissue retraction system for robot-assisted surgery is presented by Patil and Alterovitz [3]. The optimal grasping point and motion path are selected by simulating the entire space of possible paths starting from a sample of initial gripper configurations. To select grasps for unfolding a shirt, Bersch *et al.* [4] rely on the recognition of visible markers, selecting the optimal area based on prior knowledge of the markers' location on the shirt. The validity and quality of the grasp is confirmed by fitting a simplified gripper model around the 3D point cloud of the selected area.

The second class of grasping strategies involve in-hand grasping. In this case, the interaction between the hand and the object is considered with the goal of optimizing the position of each finger on the object. Generally, the fingers are not rigidly attached to the object, causing the stability of the grasp to become an important concern. This topic is well researched for the case of rigid objects, and surveys of it can be found in [5], [6]. Some simple strategies which could be expanded to planar non-rigid objects include the work of Morales *et al.* [7], [8] and Suarez *et al.* [9], who use different heuristics to compute stable, force-closure grasps in a model-free manner based on the observed contour of objects. The work of Ciocarlie and Allen [10] is also of particular interest, as it is based on optimizing the grasp in a low-dimensional subspace of possible hand configurations, greatly reducing the computation time without sacrificing the ability to generate stable grasps.

The case of non-rigid objects has not yet been widely addressed, especially in terms of selecting a grasp to optimize the in-hand shape control of the object. A few works have,

978-1-7281-1964-9/19/$31.00 ©2019 IEEE

however, explored the topic of synthesizing stable grasps. Mira *et al.* [11] studied cases where it is impossible to generate a force closure grasp without deforming the object, such as a sheet of paper or foam laying on a table. The grasp is generated by using a learning system and a database of objects and tasks with successful and unsuccessful grasps. Zaidi *et al.* [12], [13] use a model-based approach to generate a stable grasp on non-rigid objects with a multi-fingered hand. First, potential grasp polygons are fitted around a model of the object to enforce the constraints imposed by the force closure condition. Then, the interaction between the object and the hand is simulated in order to determine the forces to apply to achieve stability while causing minimal deformation.

B. Contour Shaping

The robotic task considered in this research is the shaping of the contour of a non-rigid object as it is projected on a 2D image. Works pertaining to this topic include that of Das and Sarkar [14], who use multiple autonomous manipulators to shape a large planar non-rigid object. This is achieved by minimizing an energy-like criterion to find the location of the contact points on the desired contour and mapping them to their locations on the initial contour, using the distance between the point on the current and desired contours as the control error. Berenson *et al.* [15] estimate the shape Jacobian relating the displacement of the grasp points to that of the controlled feature points by relying on the concept of diminishing rigidity. Simply stated, this is the idea that the parts of the object near a grasp point will follow its motion very closely, while areas that are further away will be less affected.

Alonso-Mora *et al.* [16] control multiple mobile manipulators carrying a large non-rigid object such as a bedsheet. A centralized planner provides general task-level guidance while delegating control to the individual robots. The object is represented as the triangulation polygon of the grasp points, such that a variety of contour shaping tasks may be defined. Recently, Wang *et al.* [17] presented a controller to reach and make initial contact with the object of interest before deforming it, based on previous works such as [18]. However, like most of the approaches described in the current literature including the above examples, the grasping point or area must be defined by the human operator.

C. Contour Tracking

It is also worthwhile to highlight a few strategies for object segmentation that are particularly appropriate for tracking the deformation of non-rigid objects. Cretu *et al.* [19] use a growing neural gas network to identify background and foreground objects in the initial frame before obtaining the mean colour of the foreground in HSV space. This allows for faster segmentation of subsequent frames by applying colour thresholding and tuning a (now non-growing) neural gas network to represent the contour.

Hui *et al.* [20] rely on a user-supplied "fixation point" near the center of the foreground object to transform the image into a log-polar representation. This causes the contour to become a mostly vertical transition between two regions, allowing the image to be processed only in the horizontal direction and simplifying the registration between successive contours. Navarro-Alarcon *et al.* [21] propose to represent the object's initial and desired contours based on a truncated Fourier series, thereby improving the robustness of the system by ignoring the noise introduced by image segmentation and hand-drawn sketches.

Overall, the most important ideas upon which this work builds are as follows. First, the principle of diminishing rigidity [15] is used to identify potential grasp points on the object contour. Then, the potential grasps are represented in a low-dimensional space to simplify the planning procedure [10]. Finally, registration between the different contours is simplified by conversion to the polar domain [20].

III. System Description

The grasp selection scheme presented in this paper relies on finding a function $D : \mathcal{P} \rightarrow \mathbb{R}$ which describes the difference between the desired contour $\mathcal{C}_d$ and the initial contour $\mathcal{C}_i$ along a finger path $P \in \mathcal{P}$, where $\mathcal{P}$ is the set of possible robotic hand finger paths. In this work, which is mostly done using image coordinates rather than an abstract mathematical description, a contour is represented as the list of pixels that correspond to the boundary of the object. In the case of the Barrett Hand, taking a 2D view which is normal to the palm of the hand allows for a convenient simplification, as the fingers trace a straight line between the edge of the object and the center of the palm, which is assumed to coincide with the center of the object. Therefore, a conversion from Cartesian to polar coordinates allows each finger path to be uniquely described by a single angle, such that $\mathcal{P} = [0°, 360°)$, discretized to the required level of precision. The rest of this section provides a step-by-step descriptions of the grasp selection procedure, initialization steps, and basic shape control loop.

A. Initialization

The experimental setup consists of a BH-262 Barrett robot hand which is installed on a table with the palm facing up. The object to deform is placed on the palm of the hand, using cardboard blocks to lift it to a height which allows the fingertips to make contact with the sides of the object. Images are acquired with a Microsoft Kinect (Xbox 360 version) mounted on a tripod and positioned to obtain a nearly vertical view of the palm of the hand. This setup is shown in Fig. 1 and allows to simulate the case where a robotic arm is used to maintain the hand at a proper height to grasp the sides of the object, while the "inverted" view simplifies contour tracking by reducing occlusions.

In order to properly define the shaping task, some user interaction is required. First, in order to reduce processing time and minimize the inclusion of background objects, the image captured from the Kinect is cropped to contain only the region of interest (the hand-object system), the bounds of which are supplied by the user. Similarly, the user must select the object

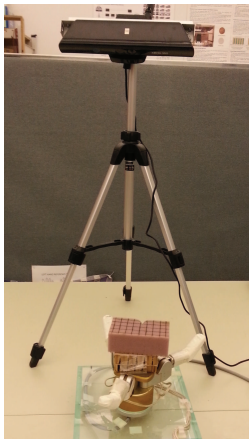


Fig. 1. Picture of the experimental setup.

to deform, which is done by selecting a point anywhere on said object.

The cropped image is then converted to the HSV colour space, and the foreground object is segmented from the rest of the scene by selecting all pixels which have a colour similar to that of the user-supplied reference point. Then, the initial object contour $\mathcal{C}_i$ and centroid (x_0, y_0) are extracted using built-in functions of standard image processing libraries (OpenCV).

The final initialization step which requires user interaction is the definition of the desired contour $\mathcal{C}_d$. This is done by drawing the desired contour on the initial frame, examples of which are shown in Fig. 5. Note that it could also be selected (with the appropriate pose adjustments) from a preexisting database of shapes or by identifying an enclosure in the scene.

B. Grasp Selection

Knowing the initial and desired contours, it is possible to build a function which describes their difference along the possible finger paths. Based on the geometry of the Barrett hand, converting the contours to polar forms $\tilde{\mathcal{C}}_i$ and $\tilde{\mathcal{C}}_d$ greatly simplifies this task. The centre of the object, (x_0, y_0), is used as the centre point of the transformation shown in (1), as it is assumed to coincide with the centre of the hand.

$$\begin{aligned}\tilde{\mathcal{C}}_i = \{(r,\theta) | r = \sqrt{(x-x_0)^2 + (y-y_0)^2}, \\ \theta = \arctan\left(\frac{y-y_0}{x-x_0}\right), (x,y) \in \mathcal{C}_i\}\end{aligned} \tag{1}$$

The sets of polar points $\tilde{\mathcal{C}}_i$ and $\tilde{\mathcal{C}}_d$ can be converted to discrete functions $C_i(\theta) = r$, taking the mapping from the corresponding (r, θ) point in $\tilde{\mathcal{C}}_i$, and $C_d(\theta) = r$, which is mapped from the (r, θ) points in $\tilde{\mathcal{C}}_d$. It may also be required to discard all points except the one with the largest r for each θ value considered as the contour may be noisy, introducing concavities and overlapping points when θ is discretized. Note that large concavities which cause the contour to break the line of sight from the center of the transformation multiple times will introduce large discontinuities, reducing the usefulness of the polar-domain contours for driving the finger motions. Similarly, r values for θs that are part of the discretized $[0, 360)$ interval $\mathcal{P}$ but that do not appear in $\tilde{\mathcal{C}}_i$ or $\tilde{\mathcal{C}}_d$ (as appropriate) are filled in by linear interpolation. Thus, it is possible to construct the contour difference function D as per (2).

$$D(\theta) = C_i(\theta) - C_d(\theta) \tag{2}$$

Based on the principle of diminishing rigidity, each contact point will be the area where the corresponding finger's influence on the shape of the object will be the greatest. It is therefore sensible to place the fingers in the areas where the difference between the initial and desired contours is the greatest. These areas are shown in the peaks of $D(\theta)$ (see Fig. 4 for an example). Since the fingers are not rigidly attached to the object, they can only apply forces towards the centre and not away from it. This matches the case where $\tilde{\mathcal{C}}_i$ has a larger r than $\tilde{\mathcal{C}}_d$ for a given θ, i.e., the positive parts of $D(\theta)$.

Given a circular indexing of $D(\theta)$ and discretization step $\delta\theta$ for $\theta \in \mathcal{P}$, the local maxima of $D(\theta)$ may be extracted as per (3). Since the contours are not smoothly defined, two passes are required to remove noise and find the true peaks. That is, in (3), $\mathcal{M}'$ is constructed to contain the local maxima of $D(\theta)$. Given the large amount of local maxima introduced by noise in the contour, $\mathcal{M}$ is built to contain only the points of $\mathcal{M}'$ with a larger Δr than their neighbours when sorted by θ.

$$\begin{aligned}\mathcal{M}' = \{(\theta, \Delta r) | \Delta r = D(\theta) \ if \\ D(\theta - \delta\theta) \leq D(\theta) > D(\theta + \delta\theta), \theta \in \mathcal{P}\} \\ \mathcal{M} = \{(\theta_i, \Delta r_i) \in \mathcal{M}' | \\ \Delta r_{i-1} \leq \Delta r_i > \Delta r_{i+1}, \theta_i < \theta_{i+1}\}\end{aligned} \tag{3}$$

Thus, $\mathcal{M}$ is the set of points which corresponds to the larger peaks of $D(\theta)$. Each point $(\theta, \Delta r) \in \mathcal{M}$ can be seen as a finger path (θ) annotated with the distance (Δr) between the initial and desired contours along this path. Applying the idea of diminishing rigidity, it desirable to place the fingers where Δr is the largest, making it a good indicator of the quality of the associated grasp path. From this, it is possible to generate $\mathcal{T}$, a set of potential 3-finger grasps, by taking all 3-combinations of $\mathcal{M}$. Given the path triplets in $\mathcal{T}$, it is necessary to find those that form valid grasps. That is, those which respect the constraints introduced by the mechanical properties of the hand as well as the requirements for grasp stability.

The Barrett hand is built with a central finger (F_3 in Fig. 2) which remains at a fixed angle while the two others (F_1 and F_2 in Fig. 2) may be rotated around the palm. However, its mechanism is such that the angles between the fingers may not be adjusted individually. In fact, the angles between the central finger and the two other fingers must be identical. Candidate path triplets are identified by the general procedure described in Algorithm 1 (illustrated in Fig. 3) which verifies that at least two of the internal angles ϕ_{ij} are identical. The path

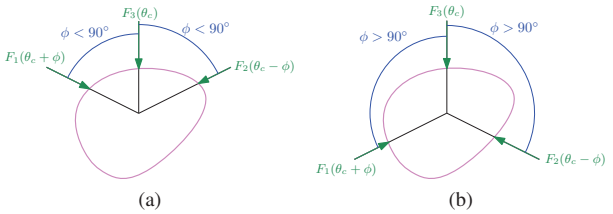


Fig. 2. Three-finger grasps respecting the Barrett hand constraints (identical angle ϕ on both sides of F_3) on an arbitrary object. The grasp in (a) is less likely to be successful than the one in (b) as it will push the object out of the hand.

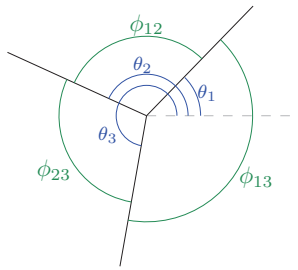


Fig. 3. Candidate finger paths in polar space represented by a triplet of angles, $(\theta_1, \theta_2, \theta_3) \in \mathcal{T}$, with black lines being interchangeably associated with the central finger, F_3. This represents a valid grasp for the Barrett hand only if at least two of the three internal angles ϕ_{ij} are identical.

Algorithm 1 Identifying valid grasps for the Barrett hand

Input: $(\theta_1, \theta_2, \theta_3) \in \mathcal{T}$ in degrees: a potential grasp, and t: a tolerance

Output: θ_c: the central angle and ϕ: the difference **if** the grasp is valid, **false** otherwise

```
θ_c = 0
φ = 0
for i = 1 to 3 do
    for j = i to 3 do
        φ_ij ← |θ_i − θ_j|
        if φ > 180° then
            φ_ij ← 360° − φ_ij
if |φ_12 − φ_13| < t then
    θ_c ← θ_1
    φ ← φ_12
else if |φ_12 − φ_23| < t then
    θ_c ← θ_2
    φ ← φ_12
else if |φ_23 − φ_13| < t then
    θ_c ← θ_3
    φ ← φ_23
else
    return false
return θ_c, φ
```

triplets which meet this condition and therefore respect the constraints of the Barrett hand form the set of valid grasps $\mathcal{G}$. For each valid grasp, the central angle θ_c and the difference ϕ between θ_c and the other angles are also recorded to support the mapping between the angles and the Barett hand's fingers in section III-C.

In terms of grasp stability, the main concern is the ability of the selected grasp to compensate all internal and external forces so that the object is not dropped. There exist a few general formulations of this constraint, namely force closure, form closure, and deform closure (see [7], [22]). In the specific case of the Barrett hand, however, a good rule-of-thumb may be derived by observing Fig. 2, which shows that grasps where $\phi > 90°$ are more likely to succeed. Therefore, those with $\phi < 90°$ are removed from $\mathcal{G}$.

Once the set of valid, stable grasps is known, all that remains is to select an appropriate one for shaping the object. Since the goal is to control the points where $D(\theta)$ is the largest, the total Δr of the grasp is used as a ranking criteria, and the grasp for which it is maximal is selected.

C. Shape Control

The selected grasp consists of three angles, namely θ_c, $\theta_c+\phi$ and $\theta_c - \phi$. These are mapped to the central 'fixed' finger of the Barrett hand and to the two 'mobile' fingers, respectively. Setting the spread to ϕ, the entire hand is physically rotated by the user (due to the absence of a wrist) so that the fingers align with the selected angles. A PID controller is then used to drive the closing of the fingers so that the selected contact points are moved from their position on $\mathcal{C}_i$ to their position on $\mathcal{C}_d$, driving the difference between the current and desired contours toward 0 along the three selected θ values. While this is unlikely to achieve $D(\theta) = 0\ \forall\theta$ due to considering only three contact points, the goal is to reach as close as possible to $D(\theta_c) \approx 0$, $D(\theta_c + \phi) \approx 0$ and $D(\theta_c - \phi) \approx 0$ within the accuracy of the contour extraction and hand alignment and under the mechanical constraints imposed by the object.

IV. Results

The proposed system was tested with a variety of non-rigid objects and desired contours, showing acceptable results given the simplicity of the implemented contour detection and control schemes. This section presents a sample of these results, namely a successful shaping operation on a soft sponge and a less successful operation on a stiff foam ball.

Fig. 4 shows the polar-domain transformation of the initial and desired contours obtained at the end of the initialisation (section III-A), as well as their difference. As described in section III-B, peaks in the difference curve show angles where the influence of the fingers should be maximized to shape the object, therefore being good candidates for the contact points. Fig. 5 shows, in order, the candidate grasp angles (i.e., θ values in $\mathcal{M}$), the selected grasp before reorienting the Barrett hand fingers, and images at different times of the manipulation sequence. They are annotated with the initial contour C_i, the desired contour C_d, as well as the contour at the current time, C_T. Fig. 6 and Fig. 7 show the control and global errors for shaping the sponge and ball, respectively. The control error for each finger is selected as the difference between the point on the current contour $C_T(\theta)$ and on the desired contour $C_d(\theta)$

for the chosen angle, while the global error is the mean square error (MSE) of the entire contour, as per (4).

$$E(\theta) = C_T(\theta) - C_d(\theta)$$
$$E_g = \frac{1}{|\mathcal{P}|} \sum_{\theta \in \mathcal{P}} E(\theta)^2 \tag{4}$$

In the case of the sponge, the finger control errors are minimized to a stable value, with fingers 1 and 3 reaching near 0. Finger 2 could not be precisely aligned with its selected grasp line, as shown in the later images of Fig. 5a. The global error is also reduced to a stable value, even though the object is not perfectly fitted to the desired contour due to the small number of control points. This is expected behaviour, as a larger number of control points should allow for more accurate shaping of the object as per [14].

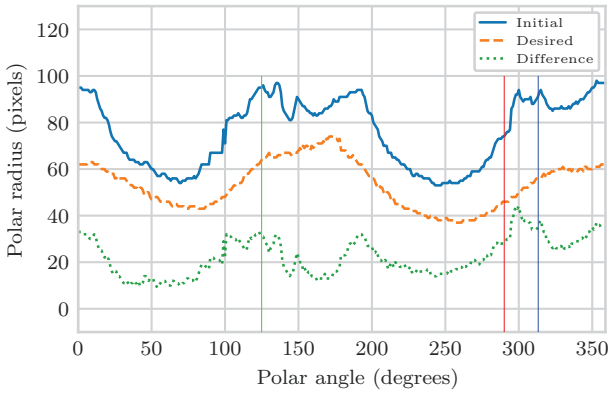
The ball was made of stiffer foam than the sponge which, combined with its spherical shape, made it more difficult to control. Nevertheless, the errors of fingers 1 and 2 were still reduced while finger 3 was not correctly aligned, causing the ball to be pushed away from the centre during the manipulation, as it can be observed in the location of the current contour with respect to the centre of projection in Fig. 5b. This caused the global error to increase as the ball was pushed away from the desired contour.

A. Time Constraints

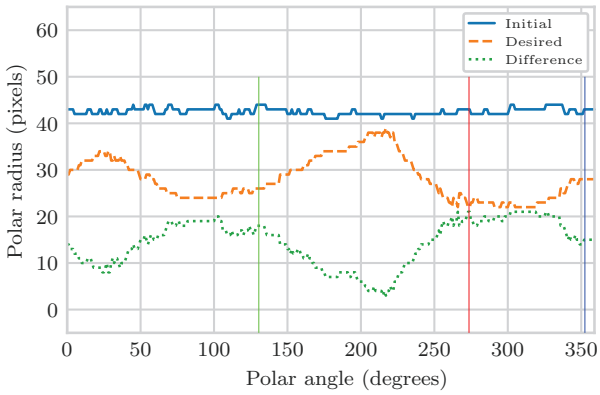
The execution time of the various selection and manipulation steps were timed informally to validate the suitability of this approach for applications which require real-time processing. Implemented in python on a 2.4 GHz processor with a few other tasks running, the grasp selection sequence required between 5 and 7 seconds of wall time depending on the number of peaks in the contour difference function, although this includes the generation of some "helper" images which are not strictly necessary for the automated selection of the grasp. Although this is hardly "real-time", it remains relatively fast, and it is expected that an optimized implementation would provide a significant reduction of this delay. On the other hand, the simple control loop implemented was able to process about 20 frames per second without moving the fingers. However, waiting for the motion to be completed between each frame requires processing only about two frames per second. There is therefore ample room for a more complex control scheme to be implemented while remaining within real-time constraints. The other steps of the algorithm, such as the initialization, selection of the desired contour, and hand alignment all required human interaction. Therefore, the measured times would be more indicative of the human's speed than of the algorithm's efficiency.

B. Limitations

The presented results highlight a few limitations in the proposed algorithm. First and foremost, it is clear that the quality of the results is extremely dependant on the quality and reliability of the contour detection algorithm, which was simplified in the current implementation. A similar observation



(a) Sponge



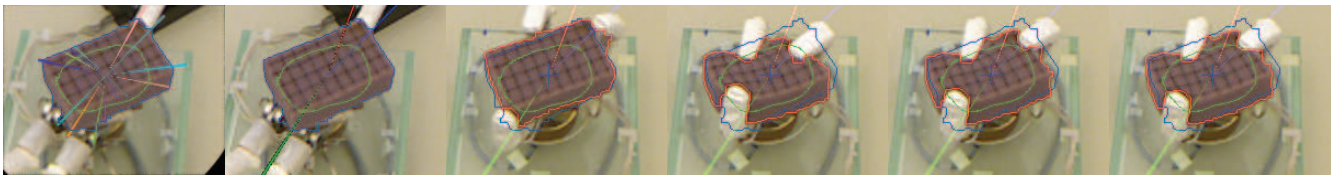
(b) Ball

Fig. 4. Polar representation of the initial contour $C_i(\theta)$, desired contour $C_d(\theta)$, as well as their difference $D(\theta)$ for both objects. Vertical lines show the approximate positions of the selected grasp points, with θ_1 in blue, θ_2 in red and $\theta_3 = \theta_c$ in green.

can be made with respect to the implemented control loop, as it is expected that an algorithm which considers the entire contour would provide better minimization of the global error that one that considers only the contact points. The case of the ball being pushed off the hand also illustrated the importance of ensuring proper force closure, which was not considered here. Finally, the use of a polar transformation to perform the registration between the different contours limits the possibilities for the definition of the desired contour, as the origin of the transformation must remain inside both contours.

V. Conclusion

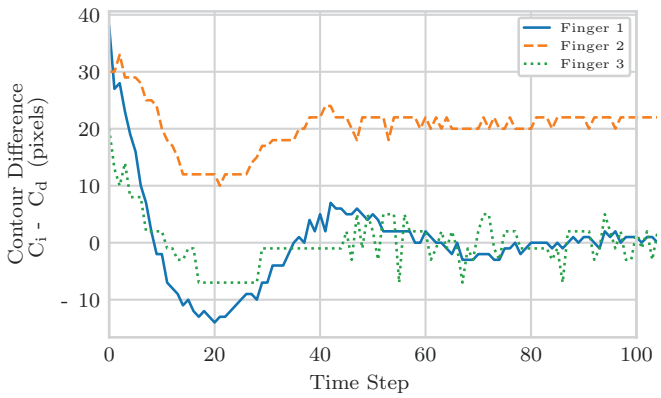
In conclusion, this work provides an original algorithm to select an optimal grasp for controlling the shape of a non-rigid object, an important task for which few automated procedures currently exist. Overall and despite some limitations, the proposed system provides a good basis for selecting grasps which aim to control the shape of a non-rigid object in addition to allowing for stable manipulation. Moreover, it does not require a prior model of the object and does not rely on computationally expensive simulations or learning systems,



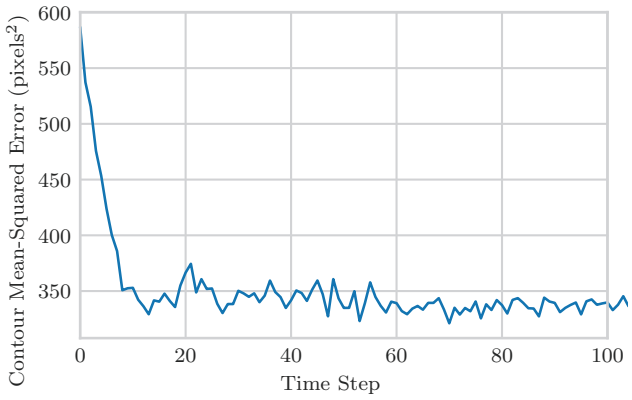
(a)

(b)

Fig. 5. Image sequences for shaping (a) the sponge, and (b) the ball. The initial contour is shown in blue, the desired contour in green and the current contour in red. Angles corresponding to peaks in $D(\theta)$ are shown in the first frame, with the length of the line being proportional to $D(\theta)$. In subsequent images, the lines show the reference angles used for the grasp with blue for θ_1 (finger 1), red for θ_2 (finger 2) and green for $\theta_3 = \theta_c$ (finger 3). The centre of the polar transformation is marked by a blue cross.

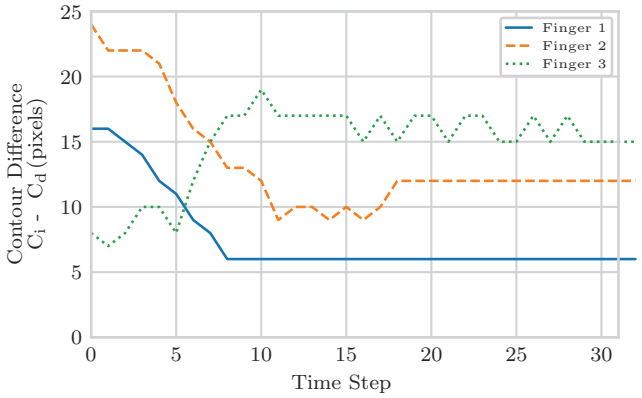


(a)

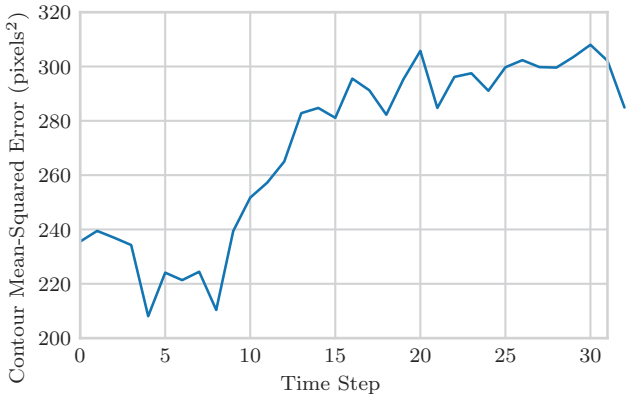


(b)

Fig. 6. Contour shaping error for the soft sponge. (a): Contour difference for the control point associated with each finger, (b): Global mean-squared error of the contour.



(a)



(b)

Fig. 7. Contour shaping error for the stiff ball. (a): Contour difference for the control point associated with each finger, (b): Global mean-squared error of the contour. The increase in global error is due to the ball being pushed off the hand.

making it suitable for implementation in robotic systems with limited computing power.

Generalizing this algorithm to different platforms requires finding a set of possible finger paths $\mathcal{P}$ based on the mechanical and geometrical properties of that platform. Then, it is possible to evaluate the contour difference along these paths to select the optimal ones.

Future work should focus on addressing the limitations of the system, namely by integrating more robust contour tracking and shape control algorithms, including a smoother contour representation [19], [21]. More accurate considerations for ensuring that the grasp achieves force closure and proper alignment of the hand are also required. Moreover, it would be of interest to experiment with a wider variety of objects, including some made of non-homogeneous materials. In addition, the variety of grasping configurations could be expanded by exploring different positions for the centre of the hand instead of constraining it to the centre of the object. Finally, this algorithm could also be expanded to handle the deformation of objects in 3D space.

ACKNOWLEDGMENT

The authors wish to acknowledge support to this research from the Natural Sciences and Engineering Research Council of Canada (NSERC), and from the Ontario Queen Elizabeth II Graduate Scholarship in Science and Technology program.

REFERENCES

[1] J. Sanchez, J.-A. Corrales, B.-C. Bouzgarrou, and Y. Mezouar, "Robotic manipulation and sensing of deformable objects in domestic and industrial applications: A survey," *The International Journal of Robotics Research*, vol. 37, no. 7, pp. 688–716, Jun. 2018. [Online]. Available: https://doi.org/10.1177/0278364918779698

[2] F. Nadon, A. Valencia, and P. Payeur, "Multi-Modal Sensing and Robotic Manipulation of Non-Rigid Objects: A Survey," *Robotics*, vol. 7, no. 4, p. 74, Nov. 2018. [Online]. Available: http://www.mdpi.com/2218-6581/7/4/74

[3] S. Patil and R. Alterovitz, "Toward automated tissue retraction in robot-assisted surgery," in *IEEE International Conference on Robotics and Automation*, May 2010, pp. 2088–2094.

[4] C. Bersch, B. Pitzer, and S. Kammel, "Bimanual robotic cloth manipulation for laundry folding," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2011, pp. 1413–1419.

[5] K. Shimoga, "Robot Grasp Synthesis Algorithms: A Survey," *The International Journal of Robotics Research*, vol. 15, no. 3, pp. 230–266, Jun. 1996. [Online]. Available: https://doi.org/10.1177/027836499601500302

[6] T. Yoshikawa, "Multifingered robot hands: Control for grasping and manipulation," *Annual Reviews in Control*, vol. 34, no. 2, pp. 199–208, Dec. 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1367578810000416

[7] A. Morales, G. Recatalá, P. J. Sanz, and A. P. Del Pobil, "Heuristic vision-based computation of planar antipodal grasps on unknown objects," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 1, 2001, pp. 583–588.

[8] A. Morales, P. J. Sanz, and A. P. del Pobil, "Vision-based computation of three-finger grasps on unknown planar objects," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002.

[9] R. Suarez, I. Vazquez, and J. M. Ramirez, "Planning four grasping points from images of planar objects," in *Proceedings of the IEEE International Symposium on Assembly and Task Planning*, 2003.

[10] M. T. Ciocarlie and P. K. Allen, "Hand Posture Subspaces for Dexterous Robotic Grasping," *The International Journal of Robotics Research*, vol. 28, no. 7, pp. 851–867, Jul. 2009. [Online]. Available: https://doi.org/10.1177/0278364909105606

[11] D. Mira, A. Delgado, C. M. Mateo, S. T. Puente, F. A. Candelas, and F. Torres, "Study of dexterous robotic grasping for deformable objects manipulation," in *23rd Mediterranean Conference on Control and Automation (MED)*, Jun. 2015, pp. 262–266.

[12] L. Zaidi, "Modélisations et stratégie de prise pour la manipulation d'objets déformables," Ph.D. dissertation, Université Blaise Pascal - Clermont-Ferrand II, Mar. 2016. [Online]. Available: https://tel.archives-ouvertes.fr/tel-01343377/document

[13] L. Zaidi, J. A. Corrales, B. C. Bouzgarrou, Y. Mezouar, and L. Sabourin, "Model-based strategy for grasping 3D deformable objects using a multi-fingered robotic hand," *Robotics and Autonomous Systems*, vol. 95, pp. 196–206, Sep. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889016308089

[14] J. Das and N. Sarkar, "Autonomous Shape Control of a Deformable Object by Multiple Manipulators," *Journal of Intelligent & Robotic Systems*, vol. 62, no. 1, pp. 3–27, Apr. 2011. [Online]. Available: http://link.springer.com/article/10.1007/s10846-010-9436-5

[15] D. Berenson, "Manipulation of deformable objects without modeling and simulating deformation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov. 2013, pp. 4525–4532.

[16] J. Alonso-Mora, R. Knepper, R. Siegwart, and D. Rus, "Local motion planning for collaborative multi-robot manipulation of deformable objects," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 5495–5502.

[17] Z. Wang, X. Li, D. Navarro-Alarcon, and Y. Liu, "A Unified Controller for Region-reaching and Deforming of Soft Objects," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2018, pp. 472–478.

[18] D. Navarro-Alarcon and Y. h Liu, "A dynamic and uncalibrated method to visually servo-control elastic deformations by fully-constrained robotic grippers," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 4457–4462.

[19] A. Cretu, E. M. Petriu, P. Payeur, and F. F. Khalil, "Deformable Object Segmentation and Contour Tracking in Image Sequences Using Unsupervised Networks," in *Canadian Conference on Computer and Robot Vision*, May 2010, pp. 277–284.

[20] F. Hui, P. Payeur, and A.-M. Cretu, "Visual Tracking of Deformation and Classification of Non-Rigid Objects with Robot Hand Probing," *Robotics*, vol. 6, no. 1, p. 5, Mar. 2017. [Online]. Available: http://www.mdpi.com/2218-6581/6/1/5

[21] D. Navarro-Alarcon and Y. H. Liu, "Fourier-Based Shape Servoing: A New Feedback Method to Actively Deform Soft Objects into Desired 2-D Image Contours," *IEEE Transactions on Robotics*, vol. 34, no. 1, pp. 272–279, Feb. 2018.

[22] K. Gopalakrishnan and K. Goldberg, "D-space and Deform Closure Grasps of Deformable Parts," *The International Journal of Robotics Research*, vol. 24, no. 11, pp. 899–910, 2005. [Online]. Available: https://journals.sagepub.com/doi/abs/10.1177/0278364905059055