Reshaping Soft Objects with a Three-Finger Robotic Hand

Tiffany Cappellari, Ting Xu, Guangzhao Yang, Xinghao Zhu
University of California, Berkeley

***Abstract*— Robotic grasping has been a popular research area for decades. More recently, people start to realize the importance of soft or deformable object grasping because of the potential applications in various areas, including bio-medical processing, the food processing industry, service robotics, robotized surgery, etc. However, not that many works focus on object reshaping. Robots that are able to reshape and mold soft objects can be a valuable asset in many different industries and workplaces. In this work, we propose a method in which a three-finger robotic hand (BarrettHand BH8-282) reshapes a soft dough into a desired shape. In the teach scenario, we want to find a mapping $\mathscr{M} : F \mapsto PC_{\text{teach}}$ where F is a provided known force, and PC_{teach} is the point cloud of the deformed object after executing the force F. In the test scenario, given a goal shape PC_{test}, we find the non-rigid transformation T from PC_{teach} to PC_{test}, and use such transformation to find the force F' such that the object will have a similar shape to PC_{test} after executing the force F'. Our approach allows us to reshape a soft object as long as the initial shapes of the object in the teach scenario and in the test scenario are the same. The project's website can be found at https://tiffanyec.github.io/206bFinalProjectWebsite/.**

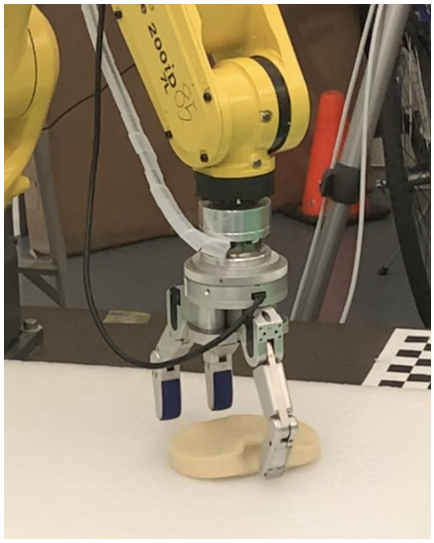

Fig. 1. BarrettHand reshaping a deformable object.

I. INTRODUCTION

Most robots today are used for rigid-body transforms and physics engines. This can prove impractical in a variety of settings because real life controls involve a variety of substances such as liquids, rigid bodies, soft bodies, and gas [1]. A robot that is able to reshape soft, malleable objects can have a variety of uses such as helping workers, ensuring efficiency in the workplace, and helping keep humans safe [2]. Some places that this technology could be applied include robotized surgery, food processing, service robots, and the medical field. Reference [3] says "research into the interaction between robotic platforms and deformable objects such as human tissue is necessary in the development of autonomous and semi-autonomous surgical systems," further supporting our belief that this technology could be used to help develop autonomous, robotized surgery along with other practical uses in the medical field. Developing a robotic system that can handle soft objects will also help open the way for topics such as sensor data processing and multi-sensor fusion, development of robust and dexterous grippers, and modeling and simulation [4]. Additionally, this function could also have more "fun" applications such as modelling clay shapes or rolling sticky rice for sushi [5].

Our idea is to have a robotic hand with three fingers reshape a soft dough into some desired shape that we ask for. We proposed to do this by splitting the task into two parts: teaching and testing. In the teaching scenario, we have the robot apply some force to the dough and then create a point map of the dough after the force has been applied to find the new shape of the object. In the testing scenario, we then tell the robot what shape we want it to reshape the dough into and based on the point clouds from the teaching scenario, the robot creates a non-rigid transformation to calculate how much force it needs to apply to the object and then executes the task. For our experiments, we used a BarrettHand BH8-282 robotic hand with 4 degrees of freedom to reshape the object, a ANUC LRMate 200iD/7L robotic arm with 6 degrees of freedom to move the robotic hand, and two Ensenso N35 cameras to create the point map of the reshaped object. We used ROS to write the code to control the robot's hand and used MATLAB to write the code to control the robot's arm and the two cameras.

In this paper, we will first reviews some recent literature and research on soft/deformable object manipulation and Gaussian Mixture Model (GMM) with Coherent Point Drift (CPD) Algorithm. Section III presents the algorithm we use in this project in order to find the non-rigid transformation between two point clouds. In section IV, we will discuss our experiment setup and analyze the results of our experiments when (i) only one force is applied, and (ii) when multiple forces are applied on the deformable object. Finally, section V summarizes the limitations and drawbacks of our approach, and things that we can improve in the future.

II. RELATED WORK

In the field of robotic grasping, most of the research has been dedicated to rigid body grasping, and only a few of them focus on soft and deformable body grasping. Prior work has discussed the problem of modeling interactions between a multi-fingered hand and a 3D deformable object for robotic grasping and manipulation tasks, using a non-linear anisotropic mass-spring system to model the mechanical behavior of the deformable object [6]. Other work uses computer vision to segment the deformations of soft objects from a video sequence, and maps them with force measurements in order to provide the necessary information to the controller of a robotic hand for safe model-based deformable object manipulation [7]. Lee et al. presents a method for learning force-based manipulation skills from demonstrations. Their method can learn a variable-impedance control strategy that trades off force and position errors, providing for the right level of compliance that applies the necessary forces at each stage of the motion [8].

[9] focus on the simultaneous control of motion and deformation for soft object manipulation. Considering viscoelastic materials, the authors use symmetric linear mass-damper-spring model for dynamical behavior analyzation. The control law is formulated using PID control for the positioned points nearest to the respective manipulated points. Alarcon et al. presents an innovative feedback representation of the objects shape based on a truncated Fourier series to guide the deformable object manipulation into a desired 2D shape instead of 3D [10]. Authors in [11] proposed to use a a forming process model to control rheological food dough. Though the controller performed well in reshaping the dough as a whole, it could not be implemented to create 3D indentations on the dough.

[12] proposed a framework to manipulate deformable linear objects. To deal with the infinite-dimension challenge, a uniform framework that includes state estimation, task planning, and trajectory planning was proposed. In this work, we used a similar framework to find a registration for the desired force. The objective of the registration is to assign correspondence between two point sets, and calculate the transformation from one set to the other. Iterative closest point (ICP, [13]) algorithm is one of most well-known methods for point set registration. The basic idea is to iteratively assign binary correspondence between two point sets by the closest distance, and find the rigid transformation by least squares to align two sets. However, ICP requires the initial positions of the two point sets to be adequately close, which is not always possible in practice.

Instead of assuming a binary correspondence based on the closest distance criterion, one-to-many relaxations have been proposed to allow for soft assignment of correspondences. Chui and Rangarajan [14] utilized GMM to calculate the correspondence by probability distribution, where one point set was treated as Gaussian centroids, and the other was the data points sampled from GMM. Moreover, they proposed to use thin plate spline (TPS, [15]) to parameterize the non-rigid transformation, leading to the well-known TPS-RPM registration algorithm [16]. However, the closed-form solution for TPS regularization does not exist for four or higher dimensions, which limits its application. Myronenko et al. [17], [18] generalized TPS to higher dimensions and proposed the CPD regularization, which can parameterize the non-rigid transformation at any dimension.

III. METHODS

Given teach initial shape, teach goal shape, teach applied force f_{teach}, test initial shape, and the test desired shape, the goal of this project was to find a force which could reshape the test initial shape to the desired shape. To do so, we proposed to first find a non-rigid transform T from the teaching goal shape to the desired goal shape, and such transform was applied to f_{teach} to find f_{test} as $f_{test} = T(f_{teach})$. Unlike rigid transformation, it's hard to formulate the non-rigid transformation as a matrix operation. Instead, a velocity field will be assigned to each point in the point cloud to represent it movement. This chapter will first describe the algorithm of non-rigid transformation, followed by the force controller we used to execute the calculated force.

A. Non-rigid Transformation

Assume the teach goal shape can be represented by N points. The nth point's position at time step t is denoted by $x_n^t \in \mathbb{R}^D$, the whole point cloud is represented by $X^t = \{x_1^t, x_2^t, \ldots, x_N^t\} \in \mathbb{R}^{N\times D}$. The point cloud of object's goal shape of at the same time step t is $Y^t = \{y_1^t, y_2^t, \ldots, y_M^t\} \in \mathbb{R}^{M\times D}$. y_m^t represents the mth point in the point cloud at time step t, M is the total point number in Y^t. Both point clouds were captured by camera and processed by filters.

In point registration step, the goal is to obtain X^t by aligning X^{t-1} to Y^t. We considered x_n^t as the nth centroid of the Gaussian Mixture Model at time step t, and Y^t is generated by sampling the GMM at time step t. Thus, the alignment can be regarded as a problem to find the centroid of the GMM.

The GMM probability density function is

$$\begin{aligned} p(y_m^t) &= \sum_{n=1}^{N} \frac{1}{N} \mathcal{N}(y_m^t; x_n^t, \sigma^2 \mathbf{I}) \\ &= \sum_{n=1}^{N} \frac{1}{N} \frac{1}{(2\pi\sigma^2)^{D/2}} \exp\left(-\frac{\|y_m^t - x_n^t\|^2}{2\sigma^2}\right) \end{aligned} \tag{1}$$

We use equal isotropic covariances $\sigma^2\mathbf{I}$ and equal membership probability $P(n) = \frac{1}{N}$ for each Gaussian component. From experiment, point cloud is not always clear, it usually contain noise point which caused by imprecise of camera and the delay of the algorithm. Thus, an addition uniform distribution is added with the weight of the uniform distribution as μ. The modified mixture model takes the form

$$p(y_m^t) = \sum_{n=1}^{N+1} p(n) p(y_m^t | n) \tag{2}$$

with

$$p(n) = \begin{cases} (1-\mu)\frac{1}{N}, & n = 1,\ldots,N \\ \mu, & n = N+1 \end{cases} \tag{3}$$

$$p(y_m^t|n) = \begin{cases} \mathcal{N}(y_m^t; x_n^t, \sigma^2\mathbf{I}), & n = 1,\ldots,N \\ \frac{1}{M}, & n = N+1 \end{cases} \tag{4}$$

We re-parametrize the GMM centroid locations by a set of parameters (x_n^t, σ^2), and estimate them by maximizing the log-likelihood function $\mathscr{L}$:

$$\begin{aligned} \mathscr{L}(x_n^t, \sigma^2|Y^t) &= \log \prod_{m=1}^{M} p(y_m^t) \\ &= \sum_{m=1}^{M} \log \left(\sum_{n=1}^{N+1} p(n) p(y_m^t|n) \right) \end{aligned} \tag{5}$$

$$(x_n^{t*}, \sigma^{2*}) = \arg\max_{x_n^t, \sigma^2} \mathscr{L}(x_n^t, \sigma^2|Y^t) \tag{6}$$

It is non-trivial to directly optimize over the log-likelihood function $\mathscr{L}$, since there is a summation inside $\log(\cdot)$ which makes convex optimization infeasible. A complete log-likelihood function Q is therefore constructed,

$$Q(x_n^t, \sigma^2) = \sum_{m=1}^{M} \sum_{n=1}^{N+1} p(n|y_m^t) \log\big(p(n) p(y_m^t|n)\big) \tag{7}$$

It can be proved by Jensen's inequality [19] that function Q is the lower bound of function $\mathscr{L}$. Therefore, increasing the value of Q will necessarily increase the value of $\mathscr{L}$ unless it is already at local optimum. Comparing the structure of Q to that of $\mathscr{L}$, the inside summation is moved to the front of $\log(\cdot)$, which provides much convenience for the following optimization.

With the definition of complete log-likelihood function, the EM algorithm [20] which runs expectation step (E-step) and maximization step (M-step) can be utilized, to iteratively estimate (x_n^t, σ^2) by maximizing Q.

With the involvement of uniform distribution, the Gaussian mixture model is able to register object nodes to reasonable locations regardless of noise and outliers. However, it is found that the estimation performance degrades dramatically if missing of point cloud happens. To achieve a higher likelihood, the mixture model will not register a single Gaussian centroid to the missing point area. Instead, those centroids will all be squeezed into the surrounding point-rich regions. Since excessive number of centroids are dividing the remaining point cloud, all estimations of the point positions become inaccurate.

To deal with this problem, [21] and [22] introduced a deactivation mechanism during registration. An independent thread will calculate the obstacle's position and check the object's visibility from camera by line projection. For those Gaussian centroids which are supposed to be invisible, they will be deactivated and kept at their original positions. Only the active centroids will take the EM update. In practice, the positions of obstacles, such as the robot arms, might be calculated by robot forward kinematics. However, in a general case, estimating the obstacle position is not always feasible and will cost a lot of computation power.

In this problem, we utilized a modified mixture model for robust state estimation under occlusion [12]. This method does not require to calculate the object's visibility. In the next subsection, a brief illustration of such model will be given.

B. Regularization by Coherent Point Drift

Under the aforementioned registration framework, there is no movement constraints on the Gaussian centroids. Each of the centroids will be inevitably registered to the point-rich area to achieve a higher likelihood value. With respect to a physical object, in contrast, there should exist an inherent topological structure that organizes all the nodes and constraints their motions in sequences.

For this reason, we introduced CPD regularization [18] to incorporate topological structure into estimation, which considers X^t to be generated by X^{t-1} by a coherent movement as a group:

$$x_n^t = x_n^{t-1} + v(x_n^{t-1}) \tag{8}$$

In (8), v is designed as a displacement function, which generates globally rigid transformation while also allows locally non-rigid deformation. We would like to find a v as smooth as possible so as to transform X^{t-1} to X^t coherently. According to the regularization theory [**?**], the function smoothness can be quantitatively measured by norm $\int_{\mathbb{R}^D} \frac{|V(s)|^2}{G(s)} ds$, where $V(s)$ is the Fourier transform of v and $G(s)$ is a symmetric filter with $G(s) \to 0$ as $s \to \infty$. The Fourier-domain norm definition basically passes v by a high-pass filter, then measure its remaining power at high frequency. Intuitively, the larger the norm, the more "oscillation" v will behave, i.e., less smoothness.

A modified likelihood function $\tilde{Q}$ is achieved by involving function v and penalizing its oscillation:

$$\begin{aligned} \tilde{Q} &= Q - \frac{\lambda}{2} \int_{\mathbb{R}^D} \frac{|V(s)|^2}{G(s)} ds \\ &= \sum_{m=1}^{M} \sum_{n=1}^{N} p(n|y_m^t) \log\left(\frac{1-\mu}{N(2\pi\sigma^2)^{D/2}}\right) \\ &\quad - \sum_{m=1}^{M} \sum_{n=1}^{N} p(n|y_m^t) \frac{\|y_m^t - x_n^{t-1} - v(x_n^{t-1})\|^2}{2\sigma^2} \\ &\quad + \sum_{m=1}^{M} p(N+1|y_m^t) \log\left(\frac{\mu}{M}\right) - \frac{\lambda}{2} \int_{\mathbb{R}^D} \frac{|V(s)|^2}{G(s)} ds \end{aligned} \tag{9}$$

The new likelihood function is parameterized by (v, σ^2), instead of (x_n^t, σ^2). $\lambda \in \mathbb{R}^+$ is a trade-off weight which balance the data fitting accuracy (from X^t to Y^t) and the topological constraints (from X^{t-1} to X^t). The negative sign before λ indicates a smaller norm, or a smoother transformation from X^{t-1} to X^t, is preferred.

Similar as the section before, the EM algorithm can be performed to estimate the parameters iteratively.

C. Execution

After we find the non-rigid transformation T and calculate the objective force F', we want to execute this force on the initial non-deformed object such that the deformed object after execution will look similar to the goal shape. We first use two Ensenso cameras to capture the initial point cloud of the non-deformed object, and find the object position in BarrettHand's base frame. Then, we move the hand right above the object and close all fingers. For finger 1 and finger 2, we implement a PID force control in order to maintain a small contact force with the object such that their impacts on the object are reduced to minimum. Finger 3 keeps pressing into the object until the magnitude of the objective force is achieved.

IV. RESULTS

During our experiments, we had two different shapes that we wanted the robot to reshape the dough into by implementing reshaping with one or multiple times. The first shape only required the robotic hand and the dough and we wanted one of the fingers to apply a force to one point on the dough and create an indentation there. The second shape used a flexible plastic mold along with the robot and the dough. By implementing reshaping four times at each side, the plastic was placed around the dough and the robots fingers pressed against the mold in order to shape the dough into a square-like shape with smooth sides.

For each of the experiments, we follow the same procedure: teaching, transformation and execution. In the following two subsections, we present the results for each of the procedures for two experiments respectively.

A. Experiment 1: Single Reshaping

In this experiment, we focus on reshaping the dough at once. Given a desired shape, the robot needs to compute a force based on the teaching scenario to execute. Figure 2. shows the initial shape, which is a round circle, for teaching scenario. After applying a force, we obtain the final shape in figure 3 that has a vector shown, indicating the force applied. Figure 4 and figure 5 present the shapes after rigid and nonrigid transformation. Finally, figure **??** shows the desired shape and actual shape achieved by the robot.

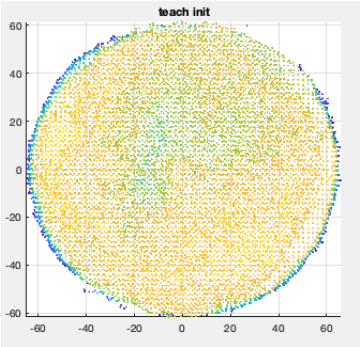

Fig. 2. Experiment 1: Initial Shape for Teaching.

plots and comparison

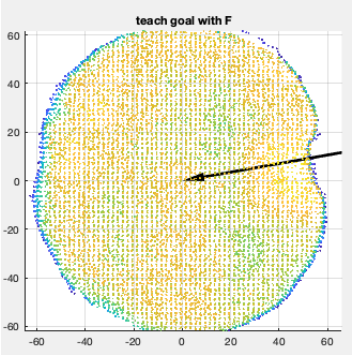

Fig. 3. Experiment 1: Final Goal Shape for Teaching.

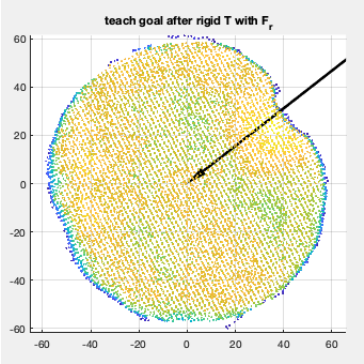

Fig. 4. Experiment 1: Rigid Transformation.

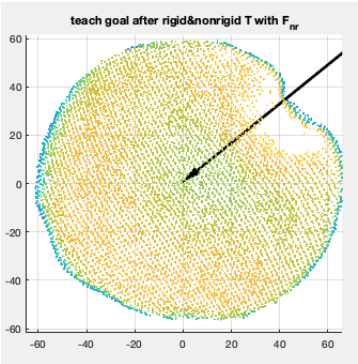

Fig. 5. Experiment 1: Non-Rigid Transformation.

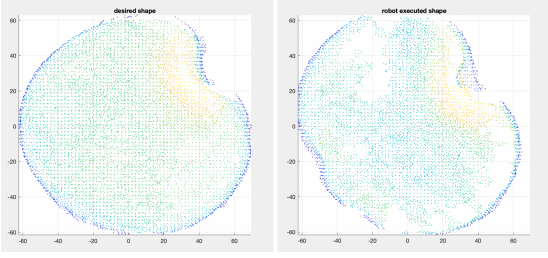

Fig. 6. Experiment 1: Comparison between Desired and Actual.

B. Experiment 2: Multiple Reshaping

In this experiment, we focus on reshaping the dough multiple times in order to achieve the desired rectangular shape. Figure 7. shows the initial shape, which is a round circle, for teaching scenario. After applying a force, we obtain the final shape in figure 8 that has a vector shown, indicating the force applied. Figure 9 and figure 10 present the shapes after rigid and nonrigid transformation. Finally, figure 11 shows the desired shape and actual shape achieved by the robot.

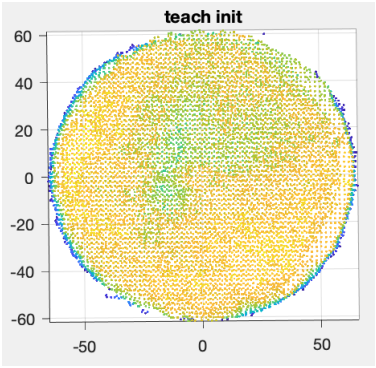

Fig. 7. Experiment 2: Initial Shape for Teaching.

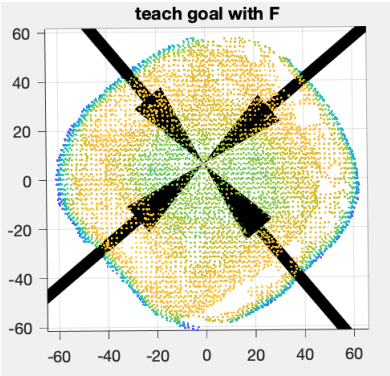

Fig. 8. Experiment 2: Initial Shape for Teaching.

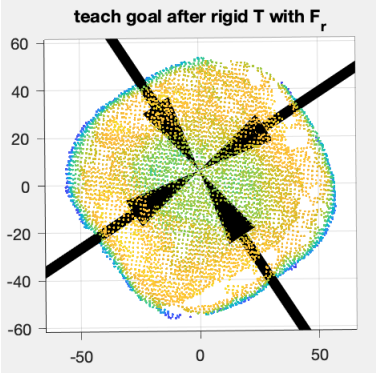

Fig. 9. Experiment 2: Rigid Transformation.

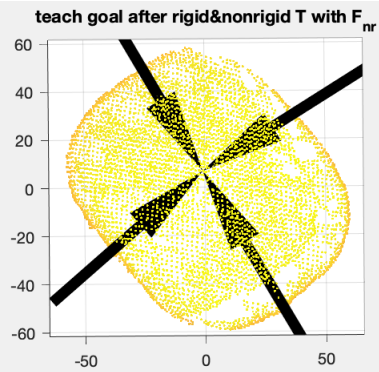

Fig. 10. Experiment 2: Non-Rigid Transformation.

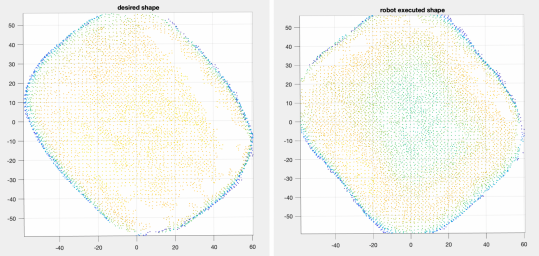

Fig. 11. Experiment 2: Comparison between Desired and Actual.

V. CONCLUSION

Overall, we were successfully able to have our robot reshape a soft dough into a desired shape with minimal errors. Our first test shape is not completely accurate because of two indentations that occur opposite of the desired indentation. This however is unavoidable because two of the robots fingers are needed to stop the object from moving while the the third finger exerts a force on the object to reshape it and the objects being soft causes the two supporting fingers to also leave imprints. We also assume the initial state of our object for the rigid body transformations and in a real world setting the initial shape of the soft object may not be guaranteed and this would not be practical. We also do not take the material that the object is made of into account. We used a dough in our experiments that is very soft and easy to reshape but other objects, such as modeling clay, can also be reshaped but is much firmer and would require much more force to be reshaped.

ACKNOWLEDGMENT

This research was supported by the EECS Department, University of California, Berkeley and the Mechanical Systems Control Lab. We thank Professor Ruzena Bajcsy, teaching assistants Chris Correa and Valmik Prabhu from the EECS Department, University of California, Berkeley and Professor Masayoshi Tomizuka from the ME Department who provided insights and expertise that greatly assisted the research.

REFERENCES

[1] Y. Li, J. Wu, R. Tedrake, J. B. Tenenbaum, and A. Torralba, "Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids," *International Conference on Learning Representations*, 2019.

[2] L. Birglen, T. Laliberté, and C. Gosselin, *Grasping vs. Manipulating*, pp. 7–31. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[3] S. Kinio, "Linear robust control in indirect deformable object manipulation," 2013.

[4] D. Henrich and H. Worn, *Robot Manipulation of Deformable Objects*, pp. 1–31. London: Springer, London, 2000.

[5] R. Matheson, "We are making robots more capable of handling objects in the real world," *World Economic Forum*, 2019.

[6] L. Zaidi, B. C. Bouzgarrou, L. Sabourin, and Y. Mezouar, "Interaction modeling in the grasping and manipulation of 3d deformable objects," *International Conference on Advanced Robotics (ICAR)*, pp. 504–509, Jul. 2015.

[7] A. Cretu, P. Payeur, and E. M. Petriu, "Soft object deformation monitoring and learning for model-based robotic hand manipulation," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, pp. 740–753, Jun. 2012.

[8] A. X. Lee, H. Lu, A. Gupta, S. Levine, and P. Abbeel, "Learning force-based manipulation of deformable objects from multiple demonstrations," pp. 177–184, May 2015.

[9] M. Shibata and S. Hirai, "Soft object manipulation by simultaneous control of motion and deformation," *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pp. 2460–2465, 2006.

[10] D. Navarro-Alarcon and Y. 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, pp. 272–279, 2018.

[11] S. Tokumoto and S. Hirai, "Deformation control of rheological food dough using a forming process model," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 2, pp. 1457–1464 vol.2, May 2002.

[12] T. Tang, C. Wang, and M. Tomizuka, "A framework for manipulating deformable linear objects by coherent point drift," *IEEE Robotics and Automation Letters*, vol. 3, pp. 3426–3433, Oct 2018.

[13] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Robotics-DL tentative*, pp. 586–606, International Society for Optics and Photonics, 1992.

[14] H. Chui and A. Rangarajan, "A feature registration framework using mixture models," in *Mathematical Methods in Biomedical Image Analysis, 2000. Proceedings. IEEE Workshop on*, pp. 190–197, IEEE, 2000.

[15] G. Wahba, *Spline models for observational data*, vol. 59. Siam, 1990.

[16] H. Chui and A. Rangarajan, "A new point matching algorithm for non-rigid registration," *Computer Vision and Image Understanding*, vol. 89, no. 2, pp. 114–141, 2003.

[17] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 12, pp. 2262–2275, 2010.

[18] A. Myronenko, X. Song, M. A. Carreira-Perpinán, *et al.*, "Non-rigid point set registration: Coherent point drift," *Advances in Neural Information Processing Systems*, vol. 19, p. 1009, 2007.

[19] M. Kuczma, *An introduction to the theory of functional equations and inequalities: Cauchy's equation and Jensen's inequality*. Springer Science & Business Media, 2009.

[20] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.

[21] J. Schulman, A. Lee, J. Ho, and P. Abbeel, "Tracking deformable objects with point clouds," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 1130–1137, IEEE, 2013.

[22] A. Petit, V. Lippiello, and B. Siciliano, "Real-time tracking of 3d elastic objects with an rgb-d sensor," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pp. 3914–3921, IEEE, 2015.