Online Shape Tracking based on Tactile Sensing and Deformation Modeling for Robot Manipulation

Jose Sanchez, Carlos M. Mateo, Juan Antonio Corrales, Belhassen-Chedli Bouzgarrou, Youcef Mezouar

***Abstract*— Precise robot manipulation of deformable objects requires an accurate and fast tracking of their shape as they deform. So far, visual sensing has been mostly used to solve this issue, but vision sensors are sensitive to occlusions, which might be inevitable when manipulating an object with robot. To address this issue, we present a modular pipeline to track the shape of a soft object in an online manner by coupling tactile sensing with a deformation model. Using a model of a tactile sensor, we compute the magnitude and location of a contact force and apply it as an external force to the deformation model. The deformation model then updates the nodal positions of a mesh that describes the shape of the deformable object. The proposed sensor model and pipeline, are evaluated using a Shadow Dexterous Hand equipped with BioTac sensors on its fingertips and an RGB-D sensor.**

I. Introduction

As interest in automating applications that require dexterous manipulation of soft objects keeps increasing, promising results have recently been achieved by applying robotic manipulation in a diversity of industries where deformable objects are commonplace. For instance, in the food industry, robots have been used to debone chickens [1] and for meat cutting [2]. The aerospace and automotive industry could reduce manufacturing costs as robots become able to manipulate cables for assembly purposes [3]. Automating surgical procedures would benefit the health industry, as in the case of robotic suturing [4]. Robot manipulation of soft objects has also domestic applications such as automated folding of clothes [5] and assisting impaired people dress themselves [6].

Despite these advances, many challenges still remain unsolved in automating tasks involving deformable objects. One such challenge, is that of manipulating the shape of a deformable object, most recently referred to as *shape servoing* [7]. To perform shape servoing on a deformable object, the robot must be able to accurately track the object's shape in order to monitor its deformation until reaching the desired shape. In this paper, we propose an approach to continuously estimate the shape of a deformable object by coupling tactile sensing with a deformable object simulator. This combination uses the local, but accurate, information provided by the tactile sensors as the input to the deformation model, which is able to estimate the actual shape of the object.

We identify our contributions as follows:

- A sensor model to estimate a contact's magnitude and location.
- A modular pipeline, combining tactile sensing with a deformation model, that tracks the shape of an object as it is deformed by a robotic hand[1].
- Quantitative evaluation of the sensor model and the deformation sensing pipeline.

Following this introduction, we review related works on deformation tracking in Section II. In Section III we describe our approach and its evaluation is presented in Section IV. The results of our evaluation are discussed in Section V and finally, our conclusions and future work are outlined in Section VI.

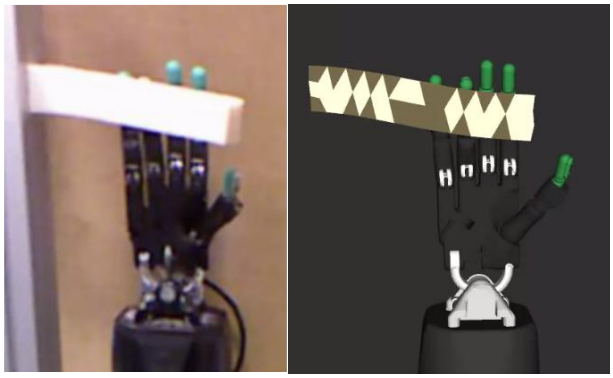


Fig. 1: Shape tracking of a deformable object.

II. Related Work

Although some approaches have considered controlling the shape of a deformable object via a robot manipulator without directly sensing its shape, they either assume that the object's shape is available [8] or use fiducial markers to extract a set of points to describe the object's shape [7]. Thus, the ability to track the shape of an object while being deformed would greatly benefit robotic manipulation tasks. In order to track the changing shape of a soft object, various approaches have been proposed using different sensing modalities such as vision and force, and by either relying on mesh models (e.g. based on the Finite Element Method (FEM) or on a Mass-Spring formulation), on mesh-free models such as Meshless Shape Matching (MSM), or not using a model at all.

Cretu et al. applied a neural gas network to the output of a vision system, to track the contour of an object while

All authors are with Université Clermont Auvergne, SIGMA Clermont, Institut Pascal BP 10448, F-63000 Clermont-Ferrand, France
Corresponding author jose.sanchez.loza@gmail.com

[1]An implementation of the code is available at https://github.com/jsanch2s/uca_deformation_sensing

a robotic hand was deforming the object. To improve the tracking accuracy, they also trained a neural network to map position and force data (obtained by joint encoders and strain gauges at the fingertips of the robotic hand) to the deformed contour as computed by the neural gas network at every timestep [9]. Although this approach does not rely on a priori information of the object (e.g. mesh model, Young Modulus and Poisson ratio), it is limited since it only provides the contour on one side of the object that is visible to the vision system.

In contrast, Tian and Jia proposed an approach that required the elasticity parameters (i.e Young modulus and Poisson ratio) to be known a priori to track the deformation of thin shell objects (e.g. a tennis ball), caused by a robotic hand, using an FEM simulation based on shell theory, where the inputs to the model were the contact forces deforming the objects [10]. Similarly, but relying on visual sensing rather than force, Petit et al. recently proposed an approach that was able to perform at 35 Hz with accurate results [11]. An approach that combined both, force and visual data, with an FEM model was described in [12]. Unlike the previous two approaches, the Young modulus and the Poisson ratio were not provided in advance and were instead estimated by minimizing the error between the observed and the simulated deformation.

Other alternatives to FEM modeling have been examined. For instance, Fugl et al. used an Euler-Bernoulli beam model to estimate the Young's modulus of a flexible bar-like object to tracks its deformation [13]. The approach required the object to be divided into sections having specific curvatures caused by the deformation. These curvatures and the object's pose, obtained via an RGB-D sensor, represented the deformation state of the beam. By having both, simulated and sensed, representations of the object's deformation, they could minimize the error between them as a function of the elasticity parameters. Mass-Spring models have also being used in combination with visual data as in the approached proposed in [14], [15]. Although approaches based on Mass-Spring models are very fast, their drawback lies on their inability to handle large deformations.

A recent approach that did not require a mesh, was described in [16]. Instead, they relied on a position-based physics simulation known as Meshless Shape Matching (MSM). As the name implies, only position information is required, which was obtained through an optical flow algorithm applied to a sequence of images. However, as the images are taken from a static camera position, the deformation can only be estimated on one side of the object.

III. Deformation Sensing Pipeline

We propose a modular pipeline able to track the shape of a soft object being deformed by a manipulator. The pipeline is composed of a sensor model (or a set of them), a force transformation component, and a deformation model, as shown in Figure 2. The approach assumes the contact forces, along with their locations, can be extracted via a sensor model (e.g. a map between a sensor's output and the contact information). The deformation model takes a tetrahedral mesh (e.g. nodes and elements connecting the nodes) describing the objects geometry and a force vector representing the forces applied to each node on a mesh; and as output it produces the new positions of the mesh's nodes. As long as this interface of the deformation model is respected, it remains interchangeable.

For this work, we use the BioTac® sensor, a biologically inspired sensor with a rigid core covered with 19 impedance electrodes. The core is wrapped by a flexible skin and the space between them is filled with a conductive liquid. As contacts deform the skin, the liquid changes its distribution, thus modifying the values of the impedance electrodes. Furthermore, the sensor is also equipped with a thermistor and pressure transducer, to produce low and high frequency sampled values of temperature and pressure [17].

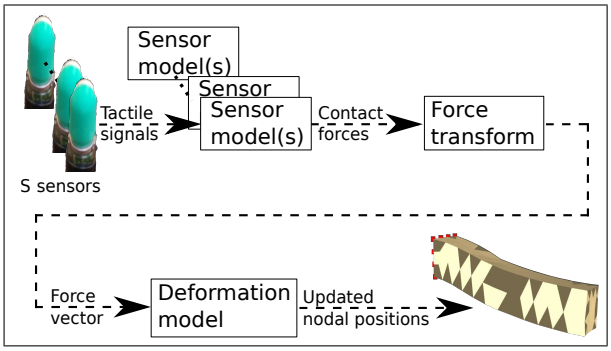


Fig. 2: Components of the deformation sensing pipeline.

A. Sensor Model

We model the BioTac sensor [17] with a sensor model that computes a single three-dimensional force caused by contacting an object as well as the contact's location.

1) Contact force magnitude estimation: Due to the complex fabrication of the BioTac tactile sensor, researchers have relied on machine learning algorithms to develop models that map the tactile signals into a three-dimensional force. These machine-learning based models have outperformed previously analytic formulations (see for instance [18]). Given that tactile signals are sequential in nature, we propose to use Recurrent Neural Networks (RNNs) to exploit the structure of the tactile data. Specifically, we use an RNN architecture called Long Short-term Memory (LSTMS) that has been successfully applied on speech recognition problems [19].

2) Contact localization: As the tactile sensor has 19 impedance electrodes distributed underneath its surface, these can be used to retrieve spatial information and thus to find the location of a contact. To localize a contact we first filter the active electrodes, e.g. the ones close enough to the contact point such that their values exceed a threshold above their resting values. We then compute the geometric centroid of the active electrodes as the mean of the positions of the m active electrodes.

$$\bar{\mathbf{x}} = \frac{1}{m}\sum_{i=1}^{m} \mathbf{p}_{\mathbf{e_i}} \tag{1}$$

where $\mathbf{p}_{e_i}$ represents the position of the i-th electrode. Once the centroid is known, it is used to find the contact point by computing the direction vectors $\mathbf{d}_i$ between the centroid and the active electrodes. In order to locate the contact point the direction vectors are multiplied by the normalized intensity of the electrodes, thus assuring that the contact point is closer to the electrodes with the highest intensities.

$$\mathbf{d}_i = (\mathbf{p}_{e_i} - \bar{\mathbf{x}})\frac{I_{e_i}}{I_e} \tag{2}$$

where I_{e_i} is the intensity value of the i-th electrode and I_e represents the sum of all active electrodes. By summing these distances we can compute the contact location c_{sensor},

$$\mathbf{c}_{sensor} = \frac{\sum_{i=1}^{m} \mathbf{d}_i}{m} + \bar{\mathbf{x}} \tag{3}$$

Finally, we project the contact point onto the finger's surface by using a geometric model of the finger. We model the finger's surface as a sphere:

$$\mathbf{c} = \mathbf{o} + \frac{r(\mathbf{c}_{sensor} - \mathbf{o})}{\|\mathbf{c}_{sensor} - \mathbf{o}\|} \tag{4}$$

where r represents the radius of the sphere (we use 7 mm) and $\mathbf{o}$ is the origin, except when the contact ($\mathbf{c}_{sensor}$ is negative on the X axis (i.e. it is in the cylindrical part of the sensor). In that case, we set $\mathbf{o} = (x, 0, 0)$ to avoid distortions (e.g. caused by using a cylindrical projection), where x is $\mathbf{c}_{sensor_x}$.

B. Force Transformation

In order to apply a force to the deformation model, the forces computed on each sensor frame must be first transformed into a common frame. The object frame is used as a reference to transform the forces

$$\mathbf{f}_i^o = \mathbf{T}_i^o \mathbf{f}_i, \tag{5}$$

where $\mathbf{T}_i^o$ is a transformation matrix relating the i-th sensor frame to the object frame. As the deformation model requires forces to be applied directly on the mesh nodes, each force needs to be distributed among the surface nodes. Since the mesh elements are tetrahedra, they can be treated as triangles on the surface and this allows for the use of a linear shape function $\mathbf{H}$ to distribute the force onto the nodes (in an inversely proportional manner based on the distance from the force's location to the three nodes).

$$\mathbf{H} = \begin{bmatrix} \frac{a_1}{a} & 0 & 0 & \frac{a_2}{a} & 0 & 0 & \frac{a_3}{a} & 0 & 0 \\ 0 & \frac{a_1}{a} & 0 & 0 & \frac{a_2}{a} & 0 & 0 & \frac{a_3}{a} & 0 \\ 0 & 0 & \frac{a_1}{a} & 0 & 0 & \frac{a_2}{a} & 0 & 0 & \frac{a_3}{a} \end{bmatrix} \tag{6}$$

$$\begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \mathbf{f}_3 \end{bmatrix} = \mathbf{H}^T \begin{bmatrix} \mathbf{f}_x^o \\ \mathbf{f}_y^o \\ \mathbf{f}_z^o \end{bmatrix} \tag{7}$$

where $\mathbf{f}_i$ is a force vector applied to node i on the X, Y and Z axes. The total area of the triangle where the force is applied is denoted by a, and a_i stands for the sub-triangle area formed between the opposite nodes of the i-node and the contact point. To select the three nodes forming the triangle where the contact point is located, we apply the k-nearest neighbors algorithm with k set equal to 3. In order to guarantee the forces are caused by contacting the object we verify that the tactile sensor frame is within a threshold distance of the closest node, in our case we used a 1 centimeter threshold.

C. Deformation Model

To compute the deformation of the object caused by external forces, the internal forces must be computed for each element in the mesh that result on new positions of the mesh's nodes. This can be achieved by solving the following differential equation:

$$\mathbf{f}_{ext} = \mathbf{M}\ddot{\mathbf{q}} + \mathbf{D}\dot{\mathbf{q}} + \mathbf{f}_{int}(\mathbf{q}) \tag{8}$$

where $\mathbf{f}_{ext}$ is the external force caused by gravity and contacts. The position, velocity and acceleration of each node n is represented by $\mathbf{q}$, $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$, respectively, with $\mathbf{q} \in \mathbb{R}^{3n}$. The mass matrix is contained in the $\mathbf{M} \in \mathbb{R}^{3n\times 3n}$ and D is the damping matrix. The internal forces are described by $\mathbf{f}_{int}(\mathbf{q}) \in \mathbb{R}^{3n}$.

We use a Co-rotational Linear FEM model [20], since it provides a balanced trade-off between accuracy and speed as it can handle large deformations and it does not require nonlinear computations. The model requires a volumetric mesh and the elasticity parameters, i.e. Young modulus and Poisson Ratio. The deformation model computes Equation 8 with initial and boundary conditions. The initial conditions are the positions and velocities of the nodes when the object is at rest, e.g. undeformed; and the boundary conditions are constrained nodes, e.g. nodes that do not change their position at any time.

IV. Experimental Evaluation

This sections describes the methods used to evaluate the accuracy of our sensor model to estimate force magnitude and contact localization; as well as the overall performance of the deformation sensing pipeline.

A. Contact force magnitude estimation

To evaluate the ability of the sensor model to estimate a three-dimensional force, we collect a dataset mapping the BioTac signals with the output of a force/torque sensor ATI Gamma[2]. To generate contacts, against the tactile sensor, with different areas shapes and sizes, we fixed probes with different tips on the force/torque sensor. We then move each finger separately, except the thumb, downwards ten times for ten seconds at nine locations on each probe (see Figure 3). This produced 228 trials, each having close to 12,000 time

[2] http://www.ati-ia.com/products/ft/ft_models.aspx?id=Gamma

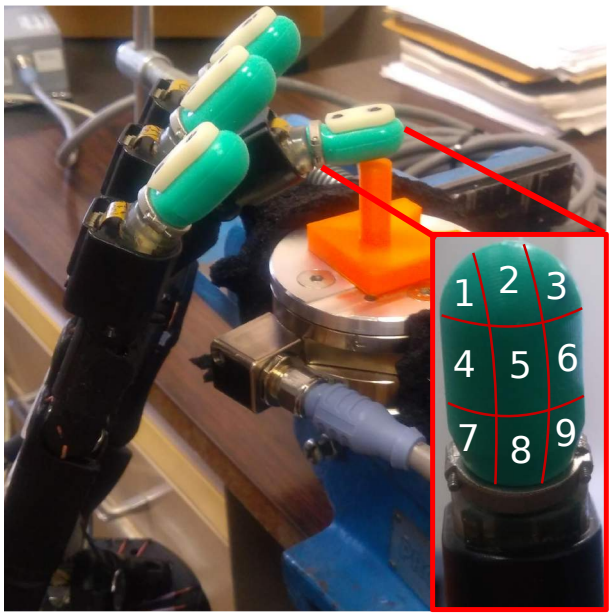


Fig. 3: Data collection for the force magnitude estimation.

steps with the input being the 19 impedance electrodes values and both, low and high, frequencies pressure values and the output being a three-dimensional force. The force values ranged between 0.1 to 1.0 N[3]. To split the training dataset from the test dataset, we used cross-validation with 80% of the original dataset as training dataset and 20% as the test dataset. As the validation dataset, we used 20% of the training dataset.

To train the RNN we used two hidden layers composed of 20 LSTM units each and the hyperbolic tangent sigmoid function as the activation function between the layers. As the optimizer, we applied Stochastic Gradient Descent with a learning rate of 0.01 in the regression layer. In order to compare the performance of our approach we use the architecture described by Su et al. in [18]. As this architecture only used the 19 impedance electrode values as inputs, we also evaluated a network with two additional vectors to consider the pressure information. Thus, we compared four architectures, namely:

1) dnn_{19} : Deep Neural Network with impedance values.
2) dnn_{21} : DNN with impedance and pressure values.
3) rnn_{19} : RNN with impedance values.
4) rnn_{21} : RNN with impedance and pressure values.

The four architectures are evaluated using the Root Mean Square Error (RMSE) and the Standardized Mean Square Error (SMSE) for the three components of the force. A comparison between the actual and estimated force magnitude is shown in Figure 4 and the evaluation results are shown in Table I.

[3]As we are concerned mainly with manipulating soft objects these are usual force values that occur when making contact with such objects.

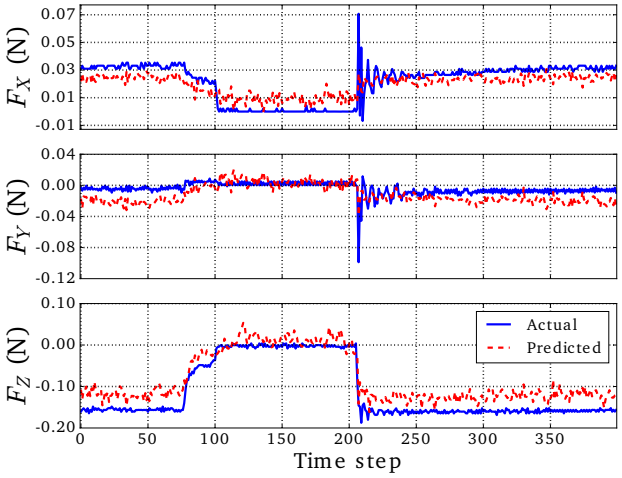


Fig. 4: Force magnitude estimation

TABLE I: Evaluation results of the force estimation.

	RMSE (in mN)			SMSE		
	F_x	F_y	F_z	F_x	F_y	F_z
dnn_{19}	41.74	94.38	344.74	1.6127	1.713	2.5225
dnn_{21}	41.95	94.59	344.71	1.6294	1.7207	2.5222
rnn_{19}	18.64	35.91	53.11	0.3213	0.2477	0.0599
rnn_{21}	18.07	31.07	51.71	0.3018	0.1854	0.0569

B. Contact localization

The contact localization algorithm was evaluated by contacting a probe five times in 12 locations distributed along the tactile sensor's fingertip. The distance between the location, as computed by our algorithm, and the fingertip was used as the error. The results are summarized in Figure 5.

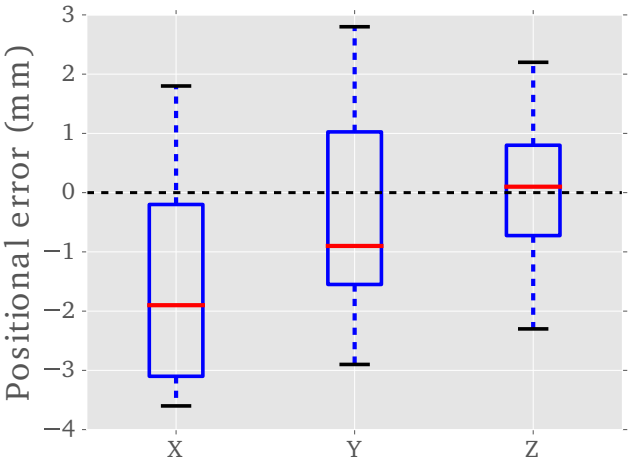


Fig. 5: Errors in the X, Y and Z axes for the contact localization algorithm.

C. Deformation estimation (Shape tracking)

To evaluate our proposed pipeline, we used the implementation of the Co-rotational Linear FEM offered by the Vega FEM library [21] and, as this is a self-contained library, we integrated it with ROS [22] to couple it with the output of our

developed sensor model. The volumetric mesh of the test objects were generated using the commercial software ANSYS and their elasticity parameters were obtained experimentally.

We used nine test objects[4] with three shapes (see Fig. 6 and Table II) and three different material properties (see Table III). The objects were then deformed using the Shadow Dexterous Hand[5]. The cube objects were grasped using two fingers, while the rest of the objects were pushed by a finger of the robot hand as they were fixed on their sides with their longest axis being horizontal, as shown in Figure 1. The sponge and bar objects start from an *undeformed* state and end in a *deformed* state. Figure 7 shows the states used for the cube objects, namely, when the object is fully visible, *unoccluded*; once contact has been made but without deformation, *occluded*; and finally, the *deformed* state.

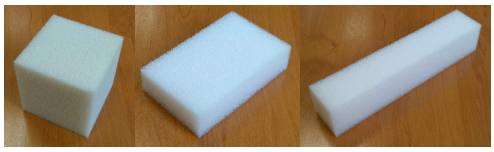


Fig. 6: Test objects: cube (hard), sponge (mid) and bar (soft).

TABLE II: Geometric information of the test objects.

	Dimensions (cm)			Mesh	
	Length	**Width**	**Height**	**Nodes**	**Elements**
Cube	6	6	6	153	486
Sponge	8	5	2	118	304
Bar	20	4	4	152	385

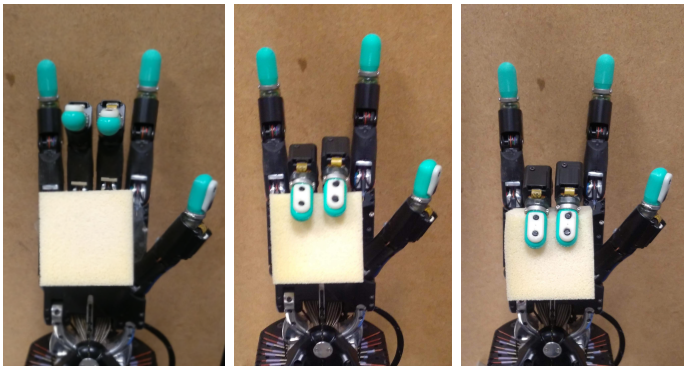
TABLE III: Material properties of the test objects.

		Elasticity parameters		
	Material name	**Mass density** (kg/m^3)	**Young modulus** (Pa)	**Poisson ratio**
Hard	HR 45	45	3800	0.15
Medium	Bultex 30	30	3200	0.15
Soft	Bultex 26	26	3000	0.15

As a ground truth measure, an uncalibrated Kinect depth sensor is used to obtain a point cloud from a visible face of the object. In order to compare the real point cloud, as obtained by the Kinect (Fig. 8a), with the mesh updated by our deformation sensing pipeline, we first render a simulated point cloud of the mesh (Fig. 8b) by placing a virtual Kinect at the same position as the original with respect to object. We then apply ray tracing to generate a point cloud of the mesh and add a Gaussian noise. To track the real object, we first segment the point cloud using the color-based segmentation proposed in [23], which uses similarity in color and spatial proximity to create clusters. A graphic user interface is then used to select the cluster that represents the object in order to track it. As the point clouds are generated without a coordinate system, we use Principal Component Analysis (PCA) to find them based on their point distributions. Since we generated a point cloud based on the mesh output of our approach, it is not possible to do a direct measure between both point clouds (e.g. Euclidean distance). To solve this, we generate an octree, based on the real point cloud with a minimum leaf size of 1 cm[6]. The accuracy of our method is then given by the ratio of points, from the simulated point cloud, that coincide with any leaf from the octree (Fig. 8c). The results are summarized in Figure 9.

[4]The objects were bought from the following vendor: `http://www.moussesurmesure.com/`

[5]`https://www.shadowrobot.com/products/dexterous-hand/`



(a) Unoccluded state. (b) Occluded state. (c) Deformed state.

Fig. 7: A cube-like object tested in the three states.

V. Discussion

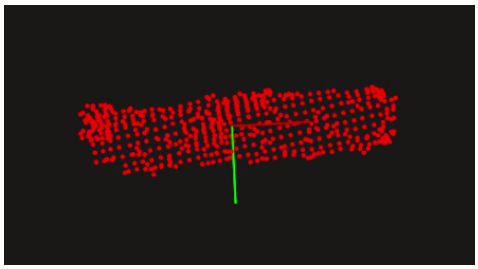
By exploiting the sequential nature of the tactile signals, RNNs proved more accurate than other deep learning algorithms in estimating the forces. Adding the pressure information did not result in significant changes. As it can be seen in Table I, the error force in the Z axis is the greatest since the data collection focused on obtaining normal forces. By only moving the fingers downwards, tangential forces were not generated as much and thus resulted in low forces values for the X and Y axes.

Figure 5 shows that the localization algorithm is able to obtain an accurate position where the error remains below five millimeters on all axes. Although, due to the location of the electrodes, the algorithm fails to localize contacts when they occur on the extreme sides.

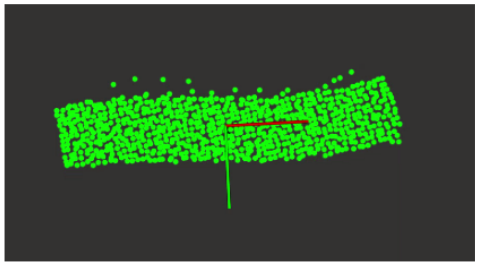
The results of the shape tracking are close to those obtained while no deformation was occurring (i.e. during the *occluded* and *undeformed* states), demonstrating the accuracy of our method. Except for the case of the sponge, where the low accuracy was due to its small size. Although the shape tracking pipeline appears to have a high variance in the accuracy results, as it can be seen in Figure 9, it is within the range of variance shown when no deformation was occurring.

A variety of other factors also contribute to errors in accuracy. For instance, the method is highly dependent on the location of the contacts. Errors in the model of the robot in simulation or a mismatch between the object's real pose and its pose in simulation have a direct impact on the performance of the pipeline. Another source of error is due

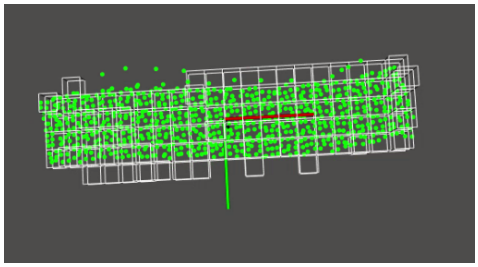
[6]We chose this size due to the accuracy of the Kinect, which is 0.5 cm.



(a) Real point cloud of the object's surface.

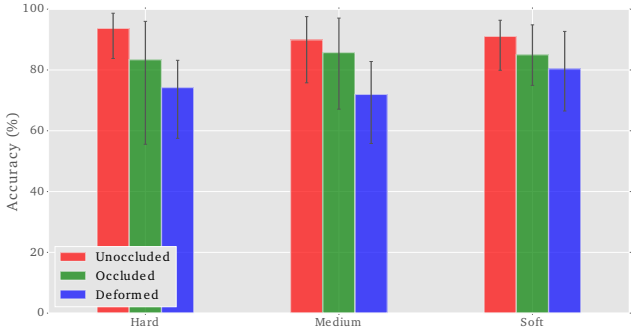


(b) Simulated point cloud on the surface of the output mesh.

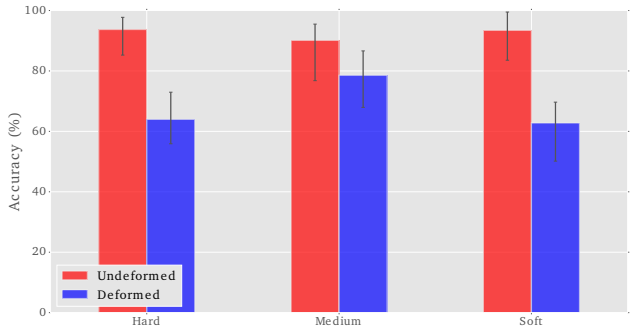


(c) The generated octree.

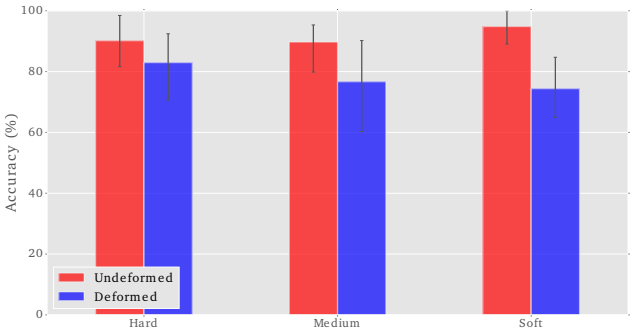
Fig. 8: Similarity evaluation of a bar-like object using RGB-D data and octrees.



(a) Accuracy results for the cube objects.



(b) Accuracy results for the sponge objects.



(c) Accuracy results for the bar objects.

Fig. 9: Evaluation results of the shape tracking.

to a lack of synchronization between the force input and the deformation model. This results in an oscillation of the mesh, that represents the object, as the deformation model might at times assume there is no external force being applied when one exists.

Other errors stem from the tactile sensors. As the contact area of the tactile sensor is relatively small, this results on the finger contacting, and subsequently deforming, the object with parts of the finger's surface that are not covered by the tactile sensor. The softness of the objects, also make it difficult for the sensor to detect contact forces.

VI. Conclusion and Future Work

We have presented a pipeline to estimate the shape of a deformable object while it is manipulated by a robotic hand. This pipeline also allows the use of multiple and different sensors by defining an interface to interact with a deformation model, namely, a sensor model. In our case, we developed and evaluated a sensor model for the BioTac sensor able to compute the three dimensional contact force along with its location. We have shown promising results with the current implementation, but certain limitations are still to be addressed. One such limitation is that of assuming contacts as points rather than areas, which causes the load distribution on the nodes to be less realistic. Also, failure to synchronize the output of the sensor model with the deformation results in an oscillatory behavior of the mesh as it computes the internal forces based on intermittent external forces. Although tactile sensors can only perceive local information, we showed that, by combining them with a deformation model, it is possible to estimate the entire shape of an object while its deformed by the tactile sensors. Tactile sensing, thus, could be a good complement to existent approaches relying on vision sensors. In our future work, besides addressing the issues mentioned above, we also intend to investigate control schemes that can be coupled with our shape estimation algorithm.

ACKNOWLEDGMENT

This work has been sponsored by the French government research program Investissements d'Avenir through the RobotEx Equipment of Excellence (ANR-10-EQPX-44) and the IMobS3 Laboratory of Excellence (ANR-10-LABX-16-01), by the European Union through the program Regional competitiveness and employment 2007-2013 (ERDF - Auvergne region) and by the Auvergne region.

REFERENCES

[1] Ai Ping Hu, Josh Bailey, Michael Matthews, Gary McMurray, and Wayne Daley. Intelligent automation of bird deboning. *2012 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*, pages 286–291, 2012.

[2] Philip Long, Wisama Khalil, and Philippe Martinet. Force/vision control for robotic cutting of soft materials. *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, (Iros):4716–4721, sep 2014.

[3] Ankit J. Shah and Julie A. Shah. Towards manipulation planning for multiple interlinked deformable linear objects. *2016 IEEE International Conference on Robotics and Automation*, pages 3908–3915, may 2016.

[4] Russell C. Jackson, Viraj Desai, Jean P. Castillo, and M. Cenk Cavusoglu. Needle-tissue interaction force state estimation for robotic surgical suturing. *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3659–3664, oct 2016.

[5] Jeremy Maitin-Shepard, Marco Cusumano-Towner, Jinna Lei, and Pieter Abbeel. Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding. *2010 IEEE International Conference on Robotics and Automation*, pages 2308–2315, 2010.

[6] Kimitoshi Yamazaki, Ryosuke Oya, Kotaro Nagahama, Kei Okada, and Masayuki Inaba. Bottom dressing by a life-sized humanoid robot provided failure detection and recovery functions. *2014 IEEE/SICE International Symposium on System Integration*, pages 564–570, dec 2014.

[7] David Navarro-Alarcon, Hiu Man Yip, Zerui Wang, Yun-Hui Liu, Fangxun Zhong, Tianxue Zhang, and Peng Li. Automatic 3-D Manipulation of Soft Objects by Robotic Arms With an Adaptive Deformation Model. *2016 IEEE Transactions on Robotics*, 32(2):429–441, apr 2016.

[8] Dmitry Berenson. Manipulation of deformable objects without modeling and simulating deformation. *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4525–4532, nov 2013.

[9] Ana Maria Cretu, Pierre Payeur, and Emil M. Petriu. Soft object deformation monitoring and learning for model-based robotic hand manipulation. *2012 IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 42(3):740–753, 2012.

[10] Jiang Tian and Yan-Bin Jia. Modeling Deformations of General Parametric Shells Grasped by a Robot Hand. *2010 IEEE Transactions on Robotics*, 26(5):837–852, oct 2010.

[11] Antoine Petit, Vincenzo Lippiello, and Bruno Siciliano. Real-time tracking of 3D elastic objects with an RGB-D sensor. *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems*, (320992):3914–3921, sep 2015.

[12] Barbara Frank, Rudiger Schmedding, Cyrill Stachniss, Matthias Teschner, and Wolfram Burgard. Learning the elasticity parameters of deformable objects with a manipulation robot. *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1877–1883, oct 2010.

[13] Andreas Rune Fugl, Andreas Jordt, Henrik Gordon Petersen, Morten Willatzen, and Reinhard Koch. Simultaneous Estimation of Material Properties and Pose for Deformable Objects from Depth and Color Images. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 7476 LNCS, pages 165–174. Springer Berlin Heidelberg, 2012.

[14] John Schulman, Alex Lee, Jonathan Ho, and Pieter Abbeel. Tracking deformable objects with point clouds. *2013 IEEE International Conference on Robotics and Automation*, (i):1130–1137, may 2013.

[15] Ibai Leizea, Hugo Alvarez, Iker Aguinaga, and Diego Borro. Real-time deformation, registration and tracking of solids based on physical simulation. *2014 IEEE International Symposium on Mixed and Augmented Reality*, pages 165–170, 2014.

[16] Püren Güler, Karl Pauwels, Alessandro Pieropan, Hedvig Kjellstrom, and Danica Kragic. Estimating the deformability of elastic materials using optical flow and position-based dynamics. *2015 IEEE-RAS 15th International Conference on Humanoid Robots*, pages 965–971, nov 2015.

[17] Nicholas Wettels, Jeremy A Fishel, and Gerald E Loeb. Multimodal Tactile Sensor. In *The Human Hand as an Inspiration for Robot Hand Development*, pages 405–429. Springer, Cham, 2014.

[18] Zhe Su, Karol Hausman, Yevgen Chebotar, Artem Molchanov, Gerald E Loeb, Gaurav S Sukhatme, and Stefan Schaal. Force estimation and slip detection/classification for grip control using a biomimetic tactile sensor. In *2015 IEEE-RAS International Conference on Humanoid Robots*, number October, pages 297–303. IEEE, nov 2015.

[19] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, number 3, pages 6645–6649. IEEE, may 2013.

[20] Matthias Müller and Markus Gross. Interactive Virtual Materials. *Proceedings of Graphics Interface 2004*, pages 239–246, 2004.

[21] F. S. Sin, D. Schroeder, and J. Barbič. Vega: Non-linear FEM deformable object simulator. *Computer Graphics Forum*, 32(1):36–48, 2013.

[22] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. ROS: an open-source Robot Operating System. *ICRA workshop on open source software*, 3(3.2):1–5, 2009.

[23] Qingming Zhan, Liang Yubin, and Yinghui Xiao. Color-Based Segmentation of Point Clouds. *Laser scanning 2009, IAPRS*, XXXVIII, P:248–252, 2009.