



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

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

► To cite this version:

Jose Sanchez, Carlos Mateo, Juan Antonio Corrales Ramón, Belhassen-Chedli Bouzgarrou, Youcef Mezouar. Online Shape Estimation based on Tactile Sensing and Deformation Modeling for Robot Manipulation. IEEE/RSJ International Conference on Intelligent Robots and Systems, Oct 2018, Madrid, Spain. 10.1109/IROS.2018.8594314 . hal-01905794

HAL Id: hal-01905794

<https://hal.science/hal-01905794>

Submitted on 26 Oct 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Online Shape Estimation 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 estimation 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 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]. A survey of further robot-based applications for manipulating deformable objects can be found in [7].

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* [8]. 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. Here, the contact information, provided by the tactile sensors, is used by a deformation model in order to estimate the shape of the object while it deforms. We identify our contributions as follows:

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

- A sensor model to estimate a contact's magnitude and location.
- A modular pipeline, combining tactile sensing with a deformation model, that estimates the shape of an object as it is deformed by a robotic hand¹.
- 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.

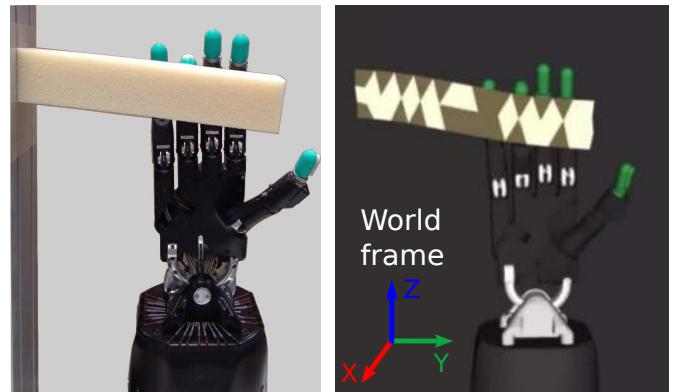


Fig. 1: Real deformation, shown on the left side, compared to the estimation of the proposed approach, visualized in RViz², shown on the right side.

II. RELATED WORK

Although some approaches have considered controlling the shape of a deformable object via a robot manipulator by either assuming that the object's shape is available [9] or by using fiducial markers to extract a set of points that describe the object's shape [8]; some real-life applications might not be feasible without directly sensing the shape of the deformable object. 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 relying on mesh

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

²<http://wiki.ros.org/rviz>

models (e.g. based on the Finite Element Method (FEM) or on a Mass-Spring formulation) or on mesh-free models such as Meshless Shape Matching (MSM). For instance, Cretu et al. applied a neural gas network to the output of a vision system, to track the contour of an object while a robotic hand was deforming the object [10]. 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 time step. 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, an approach that required the elasticity parameters (i.e Young modulus and Poisson ratio) to be known a priori, was proposed by Tian and Jia [11]. This approach used an FEM simulation based on shell theory to estimate the deformation caused by a robotic hand on thin shell objects such as a tennis ball, where the inputs to the model were the contact forces of the hand while grasping an object. 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 [12]. An approach that combined both, force and visual data, with an FEM model was described in [13]. Moreover, 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 [14]. 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 approaches proposed in [15], [16]. Although approaches based on Mass-Spring models are very fast, their drawback lies on their inaccuracy when large deformations occur. A recent approach that did not require a mesh was described in [17], where 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 estimate the shape of a soft object being deformed by a manipulator. The

pipeline is composed of a sensor model, a force transformation component, and a deformation model, as shown in Figure 2. The approach assumes that the contact forces, along with their locations, can be extracted via a sensor model. The deformation model takes a tetrahedral mesh (i.e. nodes connected by tetrahedral elements) describing the object's geometry and a force vector representing the forces applied to each node on the mesh. As output it produces the new positions of the mesh's nodes.

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 [18].

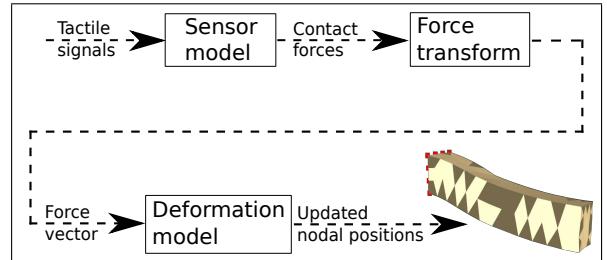


Fig. 2: Components of the deformation sensing pipeline.

A. Sensor Model

We model the BioTac sensor [18] to compute a three-dimensional force caused by contacting an object as well as the location of the contact.

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 [19]). We use Recurrent Neural Networks (RNNs) to exploit the sequential structure of the tactile signals. Specifically, we use an RNN architecture called Long Short-term Memory (LSTM) that has been successfully applied on speech recognition problems [20].

2) Contact localization: As the tactile sensor has 19 impedance electrodes distributed underneath its surface, they 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 their resting values³ (Figure 3b). We then compute the geometric centroid of the active electrodes as the mean of the positions of the m active electrodes.

³The BioTac sensor does not always initialize with the same impedance values, thus these values at rest are subtracted at each time step.

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

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.

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

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 \mathbf{c} (as shown in Figure 3c),

$$\mathbf{c} = \frac{\sum_{i=1}^m \mathbf{d}_i}{m} + \bar{\mathbf{x}} \quad (3)$$

Finally, we project the contact point onto the finger's surface by using a geometric model of the finger (Figure 3a). We model the finger's surface as a sphere:

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

where r represents the radius of the sphere (we set $r = 7$ mm) and \mathbf{o} is the origin, except when the contact \mathbf{c} 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 caused by using a spherical projection on a cylinder, where x is \mathbf{c}_x . An example of a projection when a contact lies on the spherical section is shown in Figure 3d.

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

$$\mathbf{f}_n^o = \mathbf{T}_n^o \mathbf{f}_n, \quad (5)$$

where \mathbf{T}_n^o is a transformation matrix relating the n -th sensor frame to the object frame. Furthermore, 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.

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

$$\begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \mathbf{f}_3 \end{bmatrix} = \mathbf{H}^T \begin{bmatrix} f_x^o \\ f_y^o \\ f_z^o \end{bmatrix} \quad (7)$$

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. An example of a force being distributed on the three outer nodes of a tetrahedral element is shown in Figure 4.

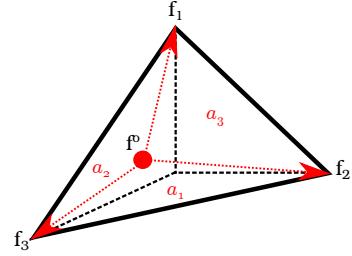


Fig. 4: Nodal distribution on a tetrahedral element.

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, internal forces must be propagated throughout the mesh to obtain the new positions of the mesh's nodes. This can be achieved by solving the following differential equation:

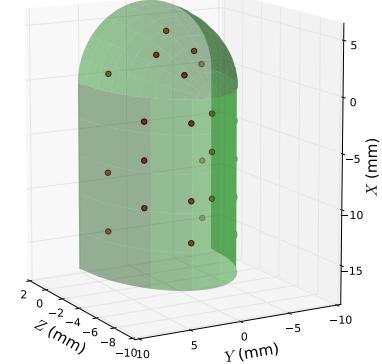
$$\mathbf{f}_{ext} = \mathbf{M} \ddot{\mathbf{q}} + \mathbf{D} \dot{\mathbf{q}} + \mathbf{f}_{int}(\mathbf{q}) \quad (8)$$

where \mathbf{f}_{ext} is the external force caused by gravity and contacts. Where the gravity vector, expressed in the world frame, is set to $\mathbf{g} = (0, 0, -9.80567 \frac{m}{s^2})$ as depicted in Figure 1. 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 \mathbf{D} is the damping matrix. The internal forces are described by $\mathbf{f}_{int}(\mathbf{q}) \in \mathbb{R}^{3n}$.

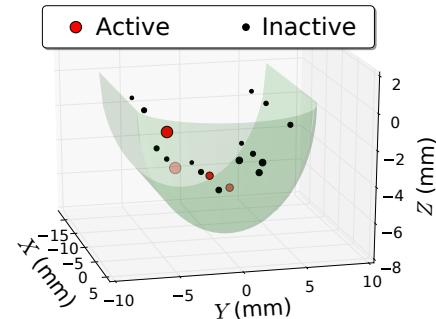
As the deformation model, we use a Co-rotational Linear FEM model [21], since it provides a balanced trade-off between accuracy and speed as it can handle large deformations and it does not require nonlinear computations. The model requires a volumetric mesh and the elasticity parameters, i.e. Young 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

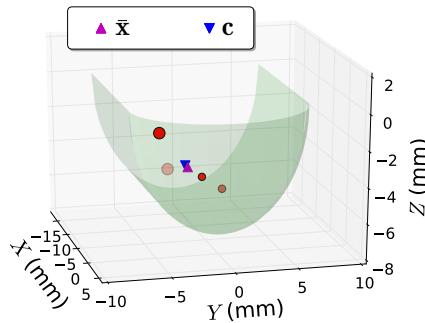
This sections describes the methods used to evaluate the accuracy of our sensor model to estimate force magnitude



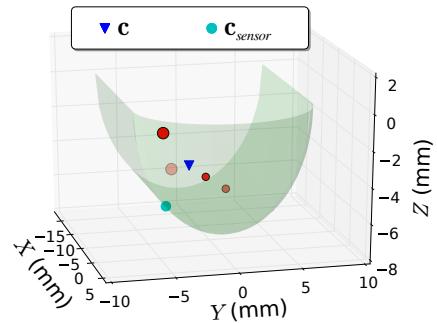
(a) Finger model with the electrodes shown in red.



(b) Active electrodes.



(c) Centroid and contact location.



(d) Projection of the contact.

Fig. 3: Finger model, shown in (a), and the steps of the contact localization algorithm: (b) thresholding of the active electrodes where the electrodes' size is shown proportional to their intensity values, (c) contact localization based on the active electrodes and their geometric centroid, (d) projection of the contact to the sensor's surface.

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 collected a dataset of the BioTac signals and the output of a force/torque sensor ATI Gamma⁴. To generate contacts on 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 5). This produced a total of 228 recordings, each having close to 12,000 time steps, containing 21 tactile signals (19 impedance and two pressure signals) and a three-dimensional force. Given that the sensors operate at different rates, the force/torque sensor has a rate of 1 KHz and the tactile sensor runs at 100 Hz, we recorded the data at the lowest rate. The force values ranged between 0.1 to 1.0 N⁵. The data was then divided by using 80% as the training dataset and 20% as the test

⁴http://www.ati-ia.com/products/ft/ft_models.aspx?id=Gamma

⁵As we are concerned mainly with manipulating soft objects these are usual force values that occur when making contact with such objects.

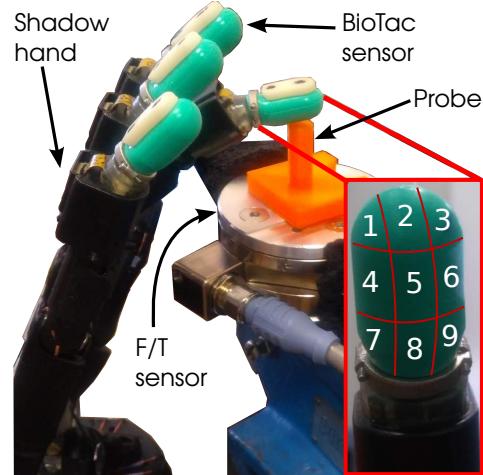


Fig. 5: Data collection for the force magnitude estimation.

dataset, and as the validation dataset, we used 20% of the training dataset.

The structure of the RNN consisted of two hidden layers, each composed of 20 LSTM units, and a fully connected output layer as shown in Figure 6. A hyperbolic tangent sigmoid function was used for the hidden layers and a linear

activation function was used for the output layer. The first hidden layer returns the full sequence (50 time steps), while the second hidden layer returns only the last time step. Thus, while the input is a sequence of tactile signals, the output is the estimated force vector at a single time step. As the optimizer, we applied Stochastic Gradient Descent with a learning rate of 0.01 in the regression layer. The RNN was implemented in Python using TFLearn [22].

In order to compare the performance of our approach we implemented a feedforward deep neural network (DNN) as described by Su et al. in [19], where the 19 impedance electrode values were used as input. Furthermore, we also evaluated networks with two additional vectors to investigate the effects of the pressure values. Thus, we compared four architectures, namely:

- 1) dnn_{19} : DNN with impedance values.
- 2) dnn_{21} : DNN with impedance and pressure values.
- 3) rnn_{19} : RNN with impedance values in 50 time steps.
- 4) rnn_{21} : RNN with impedance and pressure values in 50 time steps.

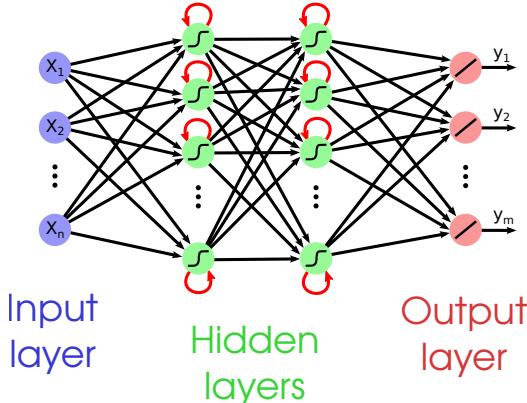


Fig. 6: RNN architecture used to estimate a three-dimensional contact force from tactile signals.

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 measured and estimated force magnitude is shown in Figure 7 and the evaluation results are shown in Table I. For a single sensor, the sensor model runs at an average speed of 185 Hz.

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, as the one shown in Figure 5, five times in 12 locations distributed along the tactile sensor's fingertip. The

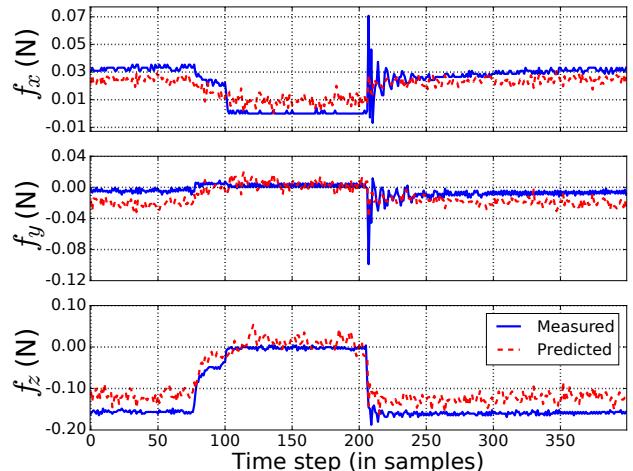


Fig. 7: Force magnitude estimation.

error was measured as the difference between the contact location, computed by the proposed algorithm, and the distance between the probe's tip and the sensor's frame, obtained by the robot's kinematics. The results are summarized in Figure 8.

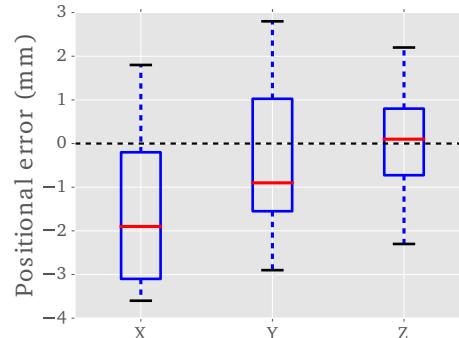


Fig. 8: Errors in the X , Y and Z axes for the contact localization algorithm.

C. Deformation sensing

To evaluate our proposed pipeline, we used the implementation of the Co-rotational Linear FEM offered by the Vega FEM library [23] and, as this is a self-contained library, we integrated it with ROS [24] 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 objects⁶ with three shapes (see Fig. 9 and Table II) and three different material properties (see Table III). To deform the objects we used the Shadow Dexterous Hand⁷. The cube objects were grasped using two fingers, while the rest of the objects were pushed by a finger

⁶The objects were bought from the following vendor: <http://www.moussesurmesure.com/>

⁷<https://www.shadowrobot.com/products/dexterous-hand/>

of the robot hand as they were fixed, using double sided tape, on their sides with their longest axis being parallel to the Y axis, as shown in Figure 1. The sponge and bar objects start from an *undeformed* state and end in a *deformed* state. Figure 10 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.

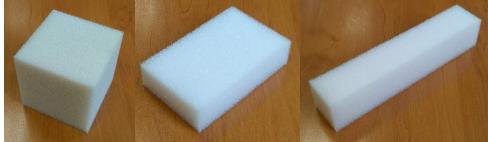


Fig. 9: Test objects: cube (hard), sponge (mid) and bar (soft).

TABLE II: Geometric information of the test objects.

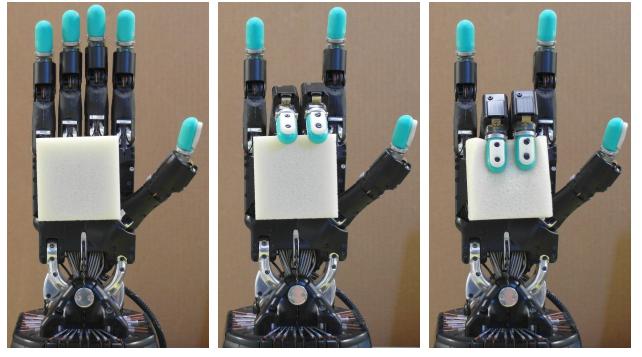
	Dimensions (cm)			Mesh	
	Length	Width	Height	Nodes	Elements
Cube	6	6	6	153	486
Sponge	8	5	2	118	304
Bar	20	4	4	152	385

TABLE III: Material properties of the test objects.

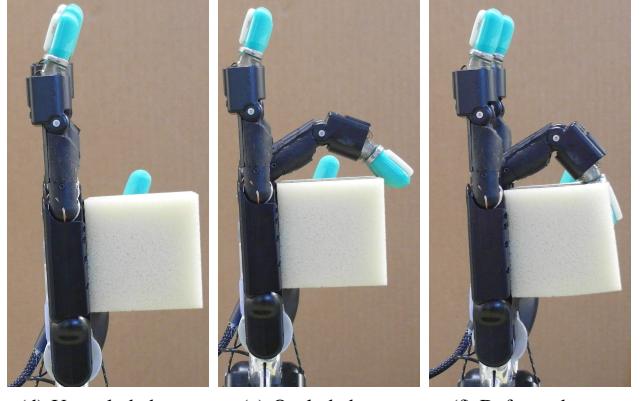
Material name	Elasticity parameters		
	Mass density (kg/m^3)	Young modulus (Pa)	Poisson ratio
Hard	HR 45	45	3800
Medium	Bultex 30	30	3200
Soft	Bultex 26	26	3000

Since there is no direct way to measure the entire shape of an object while deforming, we evaluated our approach on a single object face. To this end, we compared the similarity between two point clouds: *measured* and *simulated*. The measured point cloud was obtained by a Microsoft Kinect RGB-D sensor from the real object, as shown in Fig. 11a; while the simulated one was rendered by placing a virtual Kinect at the same position and where the real object was replaced by the mesh output of the deformation sensing pipeline. By applying ray tracing and adding Gaussian noise, the virtual Kinect generated the simulated point cloud from the mesh as it can be seen in Fig. 11b. To track the real object, we first segmented the point cloud using the color-based segmentation proposed in [25], which uses similarity in color and spatial proximity to create clusters, where the cluster representing the object to be tracked is selected manually by the user. An example of this segmentation can be seen in Figure 12. To measure the similarity of these point clouds we generated an octree based on the measured point cloud with a minimum leaf size of 1 cm⁸, where the accuracy is given by the ratio of points from the simulated point cloud that are inside the leafs from the octree (Fig. 11c). The results are summarized in Figure 13.

⁸We chose this size due to the accuracy of the Kinect, which is 0.5 cm.



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



(d) Unoccluded state. (e) Occluded state. (f) Deformed state.

Fig. 10: A cube-like object tested in the three states. Front view is shown on the top row and a side view is shown on the bottom row.

The deformation sensing component, implemented in ROS, runs at an average of 50 Hz. While the complete pipeline runs at around 20 Hz.

V. DISCUSSION

The force estimation results, shown in Figure 7, demonstrate the ability of the RNNs to accurately predict three-dimensional forces. However, the estimation of the tangential forces (f_x and f_y) is poorer than that of the normal force (f_z). This is a consequence of how the data was collected, namely, by only moving the fingers downwards tangential forces were not generated as much as the normal forces. Table I shows that the RNNs clearly outperformed the feedforward DNNs, but the addition of pressure information does not lead to a significant improvement. The contact localization algorithm was able to estimate the position of a contact with an error below five millimeters on all axes as it can be seen in Figure 8. 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 deformation estimation 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

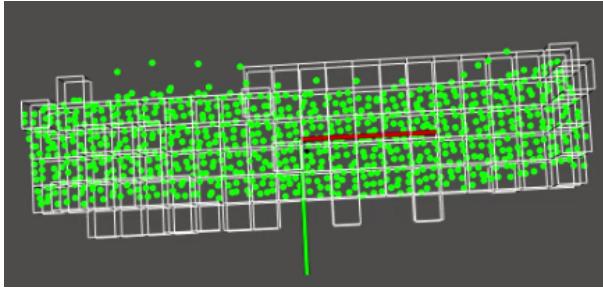
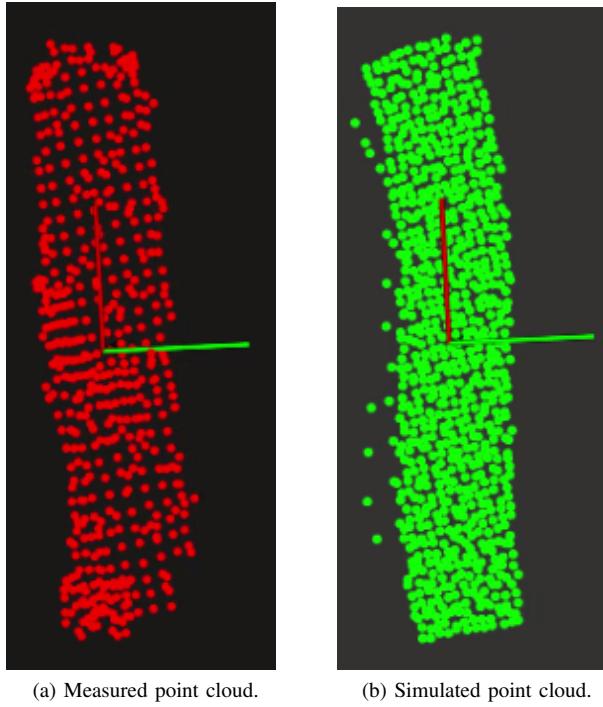


Fig. 11: Similarity evaluation of a bar-like object using RGB-D data: (a) point cloud as measured by the Kinect, (b) point cloud generated by a virtual Kinect based on the output mesh of the proposed approach, (c) octree (white) generated from the measured point cloud to measure the similarity with the simulated point cloud (green).

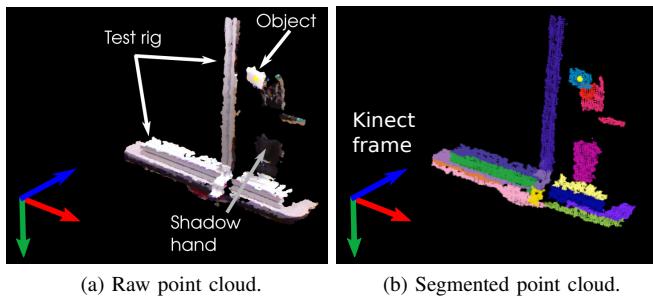
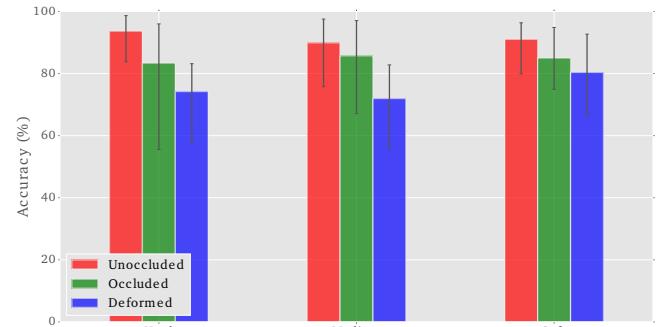
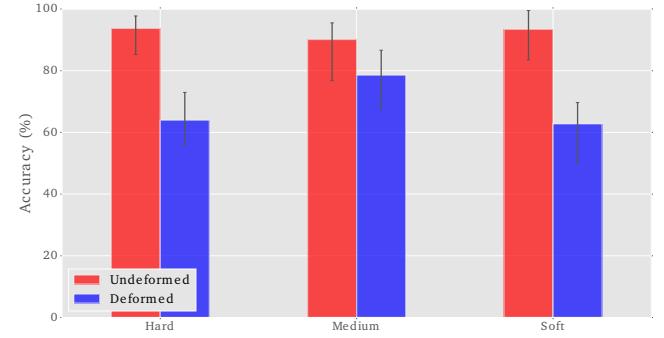


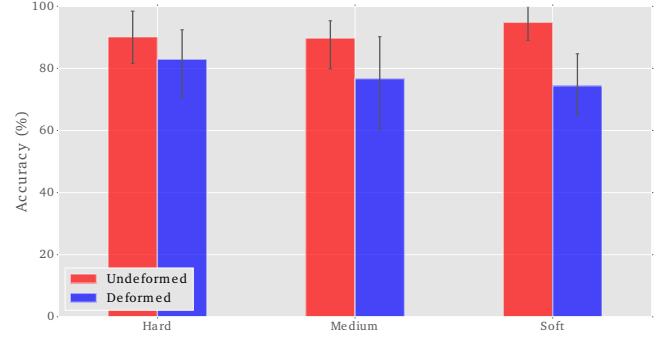
Fig. 12: Point cloud segmentation for a sponge-like object.



(a) Accuracy results for the cube objects.



(b) Accuracy results for the sponge objects.



(c) Accuracy results for the bar objects.

Fig. 13: Evaluation results of the deformation sensing.

deformation sensing pipeline appears to have a high variance in the accuracy results, as it can be seen in Figure 13, it is within the range of variance shown when no deformation was occurring. Other factors also contribute to errors in accuracy. For instance, our method is highly dependent on the location of the contacts. Thus, errors in the robot model in simulation and 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, resulting in the oscillation of the mesh, is due to intermittent contacts caused by the softness of the objects as the tactile sensors fail to detect these contacts. Other errors stemming from the tactile sensors are that they cover a small surface of the fingers is relatively small. This results on the finger contacting the object with parts of the finger's surface that are not covered by the tactile sensor. Thus, deforming the object but not having any sensor

feedback.

VI. CONCLUSION AND FUTURE WORK

We have presented a deformation sensing pipeline to estimate the shape of a deformable object while it is manipulated by a robotic hand by combining tactile sensing with a deformation model. Although tactile sensors perceive only local information we showed that, by combining them with a deformation model, it is possible to continuously estimate the global shape of an object while it is being deformed. The proposed 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 a three dimensional contact force along with its location. Although the sensor model proved accurate in estimating the force magnitude as well as the contact location, other factors not considered by the sensor model influenced the performance of the overall approach. For instance, the current sensor model assumes contacts to be points rather than areas, which causes the load distribution on the nodes to be less realistic. Also, setting contacts as boundary conditions on the mesh when the fingers touch the object is necessary to mitigate the oscillatory behavior of the simulated mesh. Nonetheless, the deformation sensing pipeline shows promising results and tactile sensing could be a good complement to existent approaches relying on vision sensors. In our future work, besides addressing the issues mentioned above, we will investigate further sensor models based on force/torque information as well as control schemes that can be coupled with our shape estimation approach.

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] A. P. Hu, J. Bailey, M. Matthews, G. McMurray, and W. Daley, “Intelligent automation of bird deboning,” in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*, 2012, pp. 286–291.
- [2] P. Long, W. Khalil, and P. Martinet, “Force/vision control for robotic cutting of soft materials,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, sep 2014, pp. 4716–4721.
- [3] A. J. Shah and J. A. Shah, “Towards manipulation planning for multiple interlinked deformable linear objects,” in *IEEE International Conference on Robotics and Automation*, may 2016, pp. 3908–3915.
- [4] R. C. Jackson, V. Desai, J. P. Castillo, and M. C. Cavusoglu, “Needle-tissue interaction force state estimation for robotic surgical suturing,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, oct 2016, pp. 3659–3664.
- [5] J. Maitin-Shepard, M. Cusumano-Towner, J. Lei, and P. Abbeel, “Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding,” in *IEEE International Conference on Robotics and Automation*, 2010, pp. 2308–2315.
- [6] K. Yamazaki, R. Oya, K. Nagahama, K. Okada, and M. Inaba, “Bottom dressing by a life-sized humanoid robot provided failure detection and recovery functions,” in *IEEE/SICE International Symposium on System Integration*, dec 2014, pp. 564–570.
- [7] 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.
- [8] D. Navarro-Alarcon, H. M. Yip, Z. Wang, Y.-H. Liu, F. Zhong, T. Zhang, and P. Li, “Automatic 3-D Manipulation of Soft Objects by Robotic Arms With an Adaptive Deformation Model,” *IEEE Transactions on Robotics*, vol. 32, no. 2, pp. 429–441, apr 2016.
- [9] 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.
- [10] A. M. 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, no. 3, pp. 740–753, 2012.
- [11] J. Tian and Y.-B. Jia, “Modeling Deformations of General Parametric Shells Grasped by a Robot Hand,” *IEEE Transactions on Robotics*, vol. 26, no. 5, pp. 837–852, oct 2010.
- [12] A. Petit, V. Lippiello, and B. Siciliano, “Real-time tracking of 3D elastic objects with an RGB-D sensor,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, no. 320992, sep 2015, pp. 3914–3921.
- [13] B. Frank, R. Schmedding, C. Stachniss, M. Teschner, and W. Burgard, “Learning the elasticity parameters of deformable objects with a manipulation robot,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, oct 2010, pp. 1877–1883.
- [14] A. R. Fugl, A. Jordt, H. G. Petersen, M. Willatzen, and R. Koch, “Simultaneous Estimation of Material Properties and Pose for Deformable Objects from Depth and Color Images,” in *Pattern Recognition*, A. Pinz, T. Pock, H. Bischof, and F. Leberl, Eds. Springer Berlin Heidelberg, 2012, vol. 7476 LNCS, pp. 165–174.
- [15] J. Schulman, A. Lee, J. Ho, and P. Abbeel, “Tracking deformable objects with point clouds,” in *IEEE International Conference on Robotics and Automation*, may 2013, pp. 1130–1137.
- [16] I. Leizea, H. Alvarez, I. Aguinaga, and D. Borro, “Real-time deformation, registration and tracking of solids based on physical simulation,” in *IEEE International Symposium on Mixed and Augmented Reality*, 2014, pp. 165–170.
- [17] P. Güler, K. Pauwels, A. Pieropan, H. Kjellstrom, and D. Kragic, “Estimating the deformability of elastic materials using optical flow and position-based dynamics,” in *IEEE-RAS International Conference on Humanoid Robots*, nov 2015, pp. 965–971.
- [18] N. Wettels, J. A. Fishel, and G. E. Loeb, “Multimodal Tactile Sensor,” in *The Human Hand as an Inspiration for Robot Hand Development*. Springer, Cham, 2014, pp. 405–429.
- [19] Z. Su, K. Hausman, Y. Chebotar, A. Molchanov, G. E. Loeb, G. S. Sukhatme, and S. Schaal, “Force estimation and slip detection/classification for grip control using a biomimetic tactile sensor,” in *IEEE-RAS International Conference on Humanoid Robots*, no. October, nov 2015, pp. 297–303.
- [20] A. Graves, A.-R. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 6645–6649.
- [21] M. Müller and M. Gross, “Interactive Virtual Materials,” *Proceedings of Graphics Interface 2004*, pp. 239–246, 2004.
- [22] D. Aymeric and Others, “TFLearn,” 2016, <https://github.com/tflearn/tflearn> (visited on 07/28/2018).
- [23] F. S. Sin, D. Schroeder, and J. Barbić, “Vega: Non-linear FEM deformable object simulator,” *Computer Graphics Forum*, vol. 32, no. 1, pp. 36–48, 2013.
- [24] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “ROS: an open-source Robot Operating System,” *ICRA workshop on open source software*, vol. 3, no. 3.2, pp. 1–5, 2009.
- [25] Q. Zhan, L. Yubin, and Y. Xiao, “Color-Based Segmentation of Point Clouds,” *Laser scanning 2009, IAPRS*, vol. XXXVIII, P, pp. 248–252, 2009.