Generation of Tactile Data from 3D Vision and Target Robotic Grasps

Brayan S. Zapata-Impata, *Member, IEEE,* Pablo Gil, *Senior Member, IEEE,* Youcef Mezouar and Fernando Torres, *Senior Member, IEEE*

***Abstract*—Tactile perception is a rich source of information for robotic grasping: it allows a robot to identify a grasped object and assess the stability of a grasp, among other things. However, the tactile sensor must come into contact with the target object in order to produce readings. As a result, tactile data can only be attained if a real contact is made. We propose to overcome this restriction by employing a method that models the behaviour of a tactile sensor using 3D vision and grasp information as a stimulus. Our system regresses the quantified tactile response that would be experienced if this grasp were performed on the object. We experiment with 16 items and 4 tactile data modalities to show that our proposal learns this task with low error.**

***Index Terms*—Robotic Perception, Tactile Feedback Estimation, Tactile Data Generation, Tactile Perception, 3D Vision**

I. Introduction

HUMANS perceive the multiple properties of objects and surfaces, like stiffness, through their sense of touch. Besides, we can estimate the physical attributes of objects by simply looking at them. That is: our visual perception allows us to approximate the feeling of touch. It is argued that our brain builds statistical models that capture visual clues which allow us to predict these properties [1]. We propose a method that will provide robots with this skill, thus enabling them to "feel" the tactile response of a grasp on an object. Our goal is that of learning to regress tactile responses using 3D visual perception and grasp information as a stimulus (see Fig. 1).

Tactile data are used for various tasks [2], like detecting the contours of an object [3], estimating its motion within the robotic hand [5] or calculating its orientation while grasped [4]. Lately, some authors have begun exploring the localisation and reconstruction of the pose of a target object using the tactile sense only [6], [7], just as a human would do when searching for an object in a box.

One of the most frequently researched areas is the classification of objects and textures [8]. When pursuing this goal, tactile data are usually combined with visual information

Manuscript received April XX, 2020; revised April XX, 2020; accepted April XX, 2020. Date of publication April XX, 2020. This paper was recommended for publication by Associate Editor XXXXX upon evaluation of the reviewer's comments. This work was supported in part by the Spanish Government and the FEDER Funds (BES-2016-078290, PRX19/00289, RTI2018-094279-B-100) and in part by the European Commission (COMMANDIA SOE2/P1/F0638), action supported by Interreg-V Sudoe.

B.S. Zapata-Impata, P. Gil and F. Torres are with the AUROVA Lab, Department of Physics, Systems Engineering and Signal Theory, University of Alicante, 03690, Spain. {brayan.impata, pablo.gil, fernando.torres}@ua.es

Y. Mezouar is with the ISPR Lab, Institut Pascal, SIGMA'Clermont, 63178, France. {youcef.mezouar}@sigma-clermont.fr

Digital Object Identifier 10.1109/TOH.2020.XXXXXXXXX

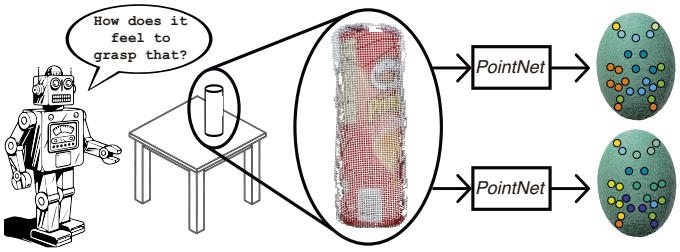

Fig. 1. We propose a method for learning to regress tactile data using 3D vision and target robotic grasps.

such as RGB images. For example, Liu *et al.* [9] paired features calculated from tactile and visual data. The authors of [10] proposed a texture classification system that learnt a joint latent space in which both modalities shared features. Remarkably, Lin *et al.* [11] framed this task differently with great results: they worked on recognising whether a pair of visual and tactile observations belonged to the same object.

Tactile perception is also frequently combined with proprioception for object recognition. Abderrahmane *et al.* [12] used an anthropomorphic hand equipped with tactile sensors on its fingertips for this task. The authors of [13] similarly presented a learning framework using a robotic hand with tactile sensors on its fingertips and palm. Luo *et al.* [14] built a dictionary of 4D points that contained the 3D coordinates of touches performed on objects and the corresponding tactile responses. Their system then recognised objects using a modified Iterative Closest Point (ICP) method.

Two other common tasks approached with tactile perception are stability prediction and slip detection. For example, Dong *et al.* [15] detected slippage by monitoring the motion of contact points using an optical tactile sensor. The authors of [16] predicted the stability of a grasp before lifting the object using deep neural networks. Abd *et al.* [17] showed that traditional signal processing methods could detect the direction of slippage. The authors of [18] showed that deep neural networks could distinguish directions of slippage by learning spatio-temporal features. More recently, such features have been learnt with soft hands and Inertial Measurement Units (IMUs) [19]. The detection of the slip direction has also been covered by combining tactile data with proprioception [20], vision [21], [22] or all of them [23].

Finally, tactile data have also been used for servoing the robot for carrying out tasks like: gently touching objects [24], discovering the actions that leverage target tactile responses [25], [26] or improving the quality of a grasp [27], [28].

Overall, the relevance of tactile perception for robotic manipulation has been demonstrated. However, the exploitation of tactile data has a serious flaw: it can be registered only during a contact. In contrast, humans can estimate the feeling of grasping an object by simply looking at it. Inspired by this, we present a novel method for learning to generate tactile responses. We provide our method with 3D point clouds and grasp data, so it learns to model the behaviour of a tactile sensor. This allows it to predict the stability of a grasp before actually making contact with the object, even in the case of using a robot without tactile sensors or in simulation.

II. Estimating Haptic Interaction from Vision: Previous Work

In recent years, Pham *et al.* have worked on estimating force from videos of humans manipulating objects. In [29], they tracked a subject's hand while manipulating a cube and estimated the object's kinematics. They used then Second-Order Cone Programming (SOCP) to calculate the force that had to be applied in order to reproduce the motion seen. Later, in [30], the authors equipped test objects with measuring tools and generated a database of videos of human hands manipulating them. They employed Recurrent Neural Networks (RNNs) to map the images and the measured kinematic features onto forces. Although these works presented low-error systems, they estimated forces for human hands actually performing a manipulation task. In contrast, we carry out our work with a robotic hand and estimate the tactile data prior to contact.

Few works in robotics literature cover the task of learning to generate tactile responses from vision. Shin *et al.* [31] worked on inferring force from videos of interactions between items and a tool attached to a servo-motor. In order to generate forces, the authors combined attention modules with Convolutional Neural Networks (CNNs) and RNNs. This work provided robust results for various objects, but the method still required the tool to make a real contact with an object in order to infer force.

Abderrahmane *et al.* [32] proposed generating tactile data from semantic descriptions of objects. They provided 19 binary haptic adjectives, which included information about the material and shape of the objects. A deconvolutional neural network was then trained with this descriptor for producing feature vectors in the tactile space. The authors improved the performance of an object recognition system using these vectors so it could handle new objects. However, it was necessary to hand-engineer the descriptor and items had to be recognised.

Recently, Hogan *et al.* [33] presented a regrasping strategy driven by synthetically generated tactile responses. The authors used the GelSight optical sensor, which registers tactile images captured by an internal camera. These images record the texture of the surface contacted. The authors used translations of these images to simulate tactile images and pair them with the movements of their gripper. They later assigned a grasp quality score to the images generated, which resulted in their system being able to find gripper adjustments that improved this score. However, this approach still needed a real grasp in order to register an initial reference tactile image.

Lee *et al.* [34] presented a cross-modal data generator using Generative Adversarial Networks (GANs): their system generated tactile data from visual perception and vice versa. They trained it using tactile images acquired from single touches with a GelSight sensor, and pictures of the scene taken with a colour camera. Two generator networks were, therefore, trained: one that produced tactile images given a picture of a piece of fabric, and another that produced the picture of the texture given the tactile image. Li *et al.* [35] trained two similar networks using a different dataset: the authors recorded tactile images and pictures of single touches on objects on a table. In this case, the networks generated a tactile image given a video sequence of the robot performing a touch on an object in the scene, or a picture of the robot and the objects given the tactile response and a reference initial picture.

These works with the GelSight provided great results in the tactile data generation task. However, this sensor produces images so these proposals are vision-to-vision models, so their task is more closely related to computer vision than to haptics. In contrast, we generate tactile data with the BioTac SP sensor, which records pressure signals. Moreover, we carry out two-fingered grasps rather than single touches. This is an important difference because the response that a tactile sensor produces from a single touch varies depending on: 1) the supporting surface of the target object and 2) its pose with respect to that support. A touch performed on the same spot on the same object can produce different tactile feedbacks if carried out on a solid table or a soft couch. It may also be different if the touch movement is perpendicular to the supporting surface or parallel to it. Grasps, however, have less variation: the same two-fingered grasp performed on the same grasping points on the same object produce similar tactile responses, independently of the supporting surface of the object.

Our contributions are, therefore, summarised as follows:

1) We present a vision-to-tactile approach for the generation of tactile data using visual perception. We use 3D point clouds since they can provide more information about objects than RGB images.
2) We propose that grasp data should also be included in the learning, thus enabling generated responses to depend on the target object, the specific area on which it would be contacted and the pose of the robotic hand.
3) We use grasps rather than single touches for the registration of tactile responses. Grasps are more useful for robotic manipulation and they are less prone to variations in the tactile feedback which depend on external factors like the environment.
4) A new dataset of 3D point clouds and real grasps is released with this work, so researchers can investigate further ways to learn to generate tactile perception.

III. Robotic System

We use the BioTac SP tactile sensor developed by SynTouch [36]. It holds 24 electrodes distributed in an internal core, which record signals from 4 emitters and measure the impedance in the fluid between them and the elastic skin of the sensor. During a contact, the greater the pressure experienced

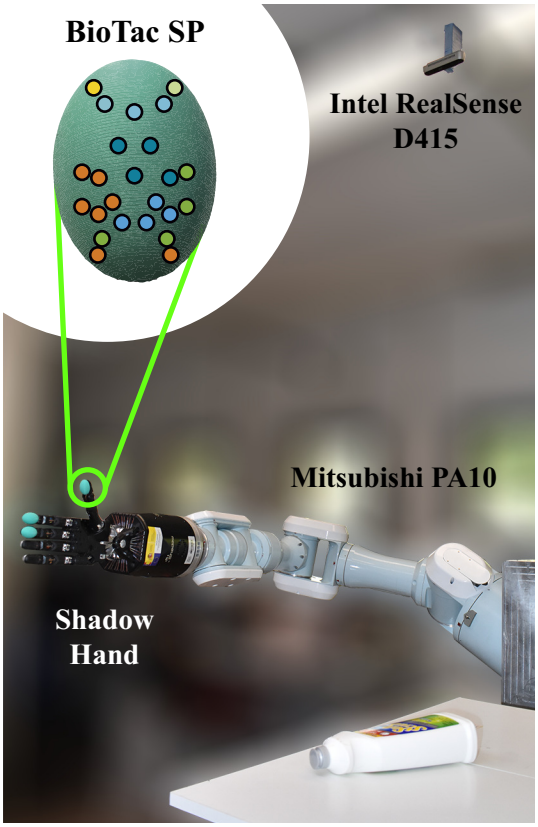

Fig. 2. BioTac SP, the tactile sensor used in this work (top-left) and rest of the robotic setup. The circles on the sensor represent the electrodes, while their colours represent different responses.

by the sensor, the lower the voltage values that are read by the electrodes. The BioTac SP also provides a global pressure measurement using a sensor in its base. The liquid inside displaces during contact, pushing itself against this pressure sensor. Although the BioTac SP measures other properties like temperature changes, we use only the electrode readings and the global pressure value in this work. Fig. 2 shows the distribution of the electrodes and the experimental setup.

We have three BioTac SP sensors on a Shadow Dexterous Hand [37], although we only use two of them in this work. More precisely, we use the sensors on the middle finger *MF* and the thumb *TH*. Hence, gripper-like grasps were performed using this multi-fingered hand. It is mounted as the end effector of a Mitsubishi PA10 robotic arm, an industrial manipulator with 7 Degrees of Freedom (DoF). The arm is mounted on a custom torso with its workspace in front of the robot, where there is a table on which we place objects. Besides, we fix in the world one Intel RealSense D415 depth camera, which captures dense 3D point clouds of the scene. The point of view of the camera provides the system with recordings from the top of the table. We present in Fig. 2 this robotic setup.

All the component in our system run on Robotic Operating System (ROS) [38], so we can read the camera stream and the BioTac SP readings. We use position controllers provided by the manufacturers to command the Shadow Hand and the PA10 arm. Finally, trajectories are generated using MoveIt! [39].

IV. Methodology

Our target task is to generate tactile readings that would be registered by the BioTac SP sensors if a grasp were executed on an object. We propose to train a supervised network with 3D point clouds of objects, desired grasp configurations and target tactile responses. We identify three key questions: A) how should we represent the object and the grasp?; B) what should the output of the network be?, and C) what should the architecture of such a network be? This section covers our answers to these questions, which define our proposal. We also describe the dataset of real grasps collected for experimentation.

A. Visual Representation

We propose 3D point clouds for representing the object. This type of structure represents the geometry of the object better than a 2D image, which should be useful as regards modelling the tactile response: areas with similar geometrical shapes should produce similar forces on the tactile sensor under similar grasp conditions. We segment the objects so that clouds contain only the points that belong to them. Moreover, we work only with 3D coordinates and colour channels. In experimentation (Section V), we investigate the effects of these features on our learning system. Note that we use a single RGBD camera, so the point clouds contain a partial view of the object. We worked under this constraint because it covers a wide range of scenarios, like settings in which items are in a wardrobe and capturing more views is not possible.

Using only a visual representation of the object might not be sufficiently informative to allow the network to generate accurate tactile responses. We propose to alleviate this by including grasp data. We experiment with two grasp representations: 1) a coarse-grained representation $\Theta_C = \{g_1, g_2\}$ that contains the 3D coordinates of the two grasping points g_1, g_2 that would be contacted, and 2) a fine-grained representation $\Theta_F = \{g_1, g_2, R\}$ that also contains a rotation matrix R that represents the pose of the wrist during the desired grasp. We do not include the translation to the wrist because it is a robot dependant value and it had no effects on the performance of the system in early experiments.

B. Tactile Response

Two levels of granularity are identified for this task: in the coarse version, the system has to learn to regress the global pressure value, called *DC pressure* or PDC, which is obtained from the pressure sensor on the base of the BioTac SP. There is one value for each sensor: PDC_{mf} and PDC_{th}. In the fine-grained problem, the system has to learn to regress the electrodes readings, denoted as E in this work. There are 24 values for each sensor: $E_{mf} = \{e_1, e_2, ..., e_{24}\}$ and $E_{th} = \{e_1, e_2, ..., e_{24}\}$.

The values provided by the BioTac SP are in custom discrete units. Although it does not produce force values in Newtons, there is a proved relationship between these custom units and the force experienced [40]. We, therefore, work with these custom units directly, taking into account that higher forces mean

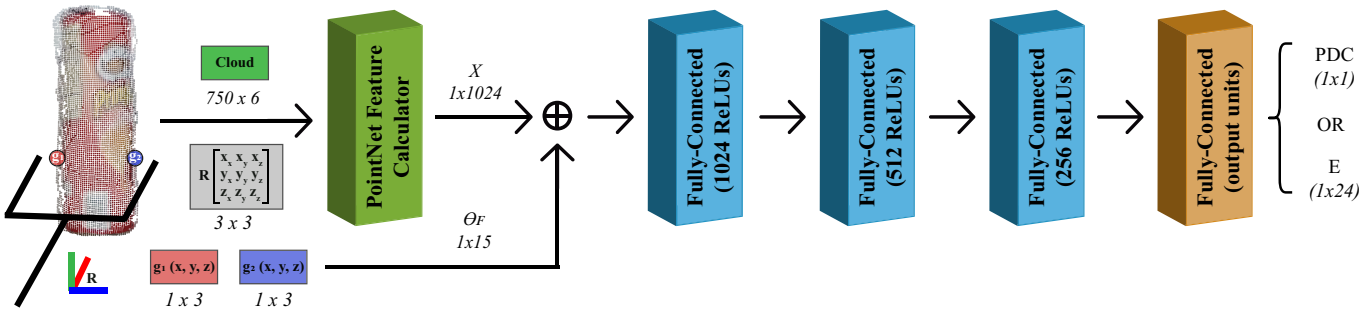

Fig. 3. Architecture proposed as regards learning to regress tactile responses based on PointNet and fully connected layers. The same architecture can be used to learn to generate single pressure values (PDC) or electrode (E) activations for one tactile sensor.

that the read values are closer to 0. In addition, the ranges of values provided by each of the two data modalities – PDC and E – are different. The discrete ranges of values found empirically for these tactile modalities and our sensors are specifically: $PDC_{th} \in [2500, 3400]$, $PDC_{mf} \in [1600, 2300]$, $E_{th} \in [100, 3600]$ and $E_{mf} \in [500, 3800]$. Each sensor has a slightly different sensitivity to contact owing to differences in the amount of liquid inside the sensor itself. The sensors consequently register readings with different ranges, even for the same source of data.

C. Neural Learning

Our task is to generate the tactile responses that would be registered by the BioTac SP sensor during a grasp of an item without actually executing the grasp. For this purpose, we have proposed an input representation made of a 3D point cloud and a description of the grasp. As regards the output, we have proposed PDC and E tactile modalities. In this section, we describe the system that maps our inputs onto these outputs.

We propose a neural network based on PointNet [41] to learn this task. We choose this network because it has performed well in related tasks. PointNet processes point clouds in which each point has a set of features, like 3D coordinates and colour channels. Convolutions are employed to calculate a vector X with 1024 features, which represents the input point cloud for the target task. We use these features as input for a set of Fully-Connected (FC) layers with Rectified Linear Units (ReLUs), with the exception of the last one, which provides the regression results, as shown in Fig. 3.

In order to include grasp data, we concatenate extra values to the X feature vector depending on the grasp representation: 6 values are used for Θ_C which are the 3D coordinates of the two grasping points, while 15 values are used for Θ_F, which are those 6 coordinates plus 9 values for the axes of the rotation matrix R. As a result, the first fully-connected layer after the PointNet feature calculator receives a vector with either 1030 or 1039 values depending on the grasp representation. An example of an architecture trained with Θ_F is shown in Fig. 3. Experiments with more variations of this architecture are shown in Section V.

D. Data Transformation

Our dataset, described in Section IV-E, holds clouds with sizes that range from 814 to 6525 points. However, every cloud must be of the same size in order to train the PointNet feature calculator. Thereby, we downsample our clouds. As a side benefit, processing smaller clouds requires less computation power and speeds up the training, although it may result in a loss of information. This is dealt with by sampling points uniformly without replacements online, such that every time a cloud is seen during training, its sampled version is different. We have experimented with various target sizes, which are discussed in Section V.

We also experiment with three normalisation techniques in order to process the input of the learning system:

1) *Unit Sphere Normalisation.* We normalise point clouds to the unit sphere whose centre is at the cloud's centroid. That is, if c is the centroid of the cloud $\mathbb{C}$, we transform the reference frame of every point $p \in \mathbb{C}$ by simply subtracting the centroid to it: $p_t = p - c$, where p_t is the transformed version of p. As a result, the centroid c becomes the origin of the new reference frame. We then scale points to the range $[0, 1]$ using the Euclidean distance ϕ to the furthest point p from the centroid c, so the normalised point is $p_n = p_t/\phi$. Grasp data is also normalised using this method, for example: $g_{in} = (g_i - c)/\phi$, where g_{in} is the i-th grasping point normalised.
2) *Min-Max Scaling.* Each feature of the cloud and the grasp data is scaled using the minimum and maximum values of the features of the same sample: $p_n = (p - min(\mathbb{C}))/(max(\mathbb{C}) - min(\mathbb{C}))$. We then scale each feature to the range $[-1, 1]$.
3) *Transform to Wrist Frame.* We use the rotation matrix R from the grasp data and an empty translation vector in order to form a transformation matrix. This transformation matrix is then applied to all the points in the cloud and to the grasping points. As a result, the Cartesian coordinates in our samples are defined with respect to a reference frame whose centre is the camera, but rotated as regards the orientation of the wrist. Since this normalisation does not affect colour, we scale this channel to range $[-1, 1]$.

As mentioned in Section IV-B, our tactile modalities are in different ranges of discrete values. In order to improve the convergence of the learning method, we scale them as well. This is done by applying the *Min-Max Scaling* to these data sources using the ranges described in Section IV-B.

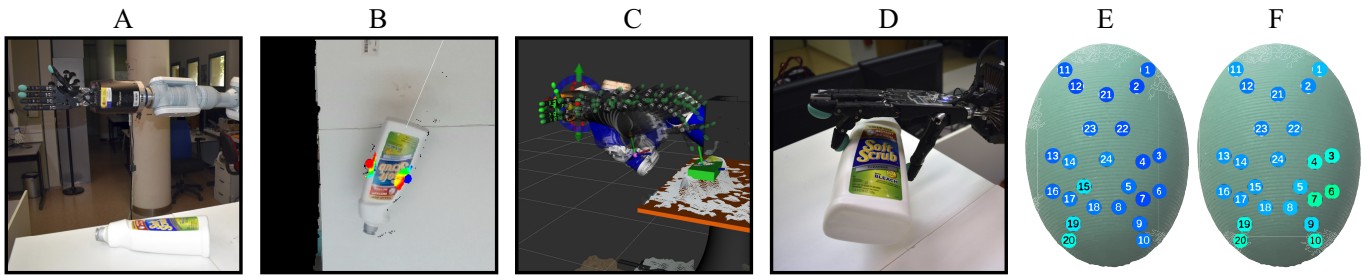

Fig. 4. Some of the steps in the sequence followed in order to collect data: (A) preparing the robot, (B) recording a point cloud and generating a grasp, (C) calculating a trajectory, (D) grasping the actual object, (E) E_{mf} and (F) E_{th} responses of electrodes for both sensors, where colour represents pressure.

Finally, we explore a data augmentation technique in order to potentially increase the variability of inputs. This is done by rotating point clouds and grasps using random angles over their axes. The resulting values of the coordinates are different, but the samples maintained their spatial relationships. We define a rotation matrix for each axis: $R_{\vec{x}}$ uses an angle α in order to rotate around the $\vec{x}$ axis, $R_{\vec{y}}$ uses an angle β in order to rotate around $\vec{y}$, and $R_{\vec{z}}$ uses the angle γ in order to rotate around $\vec{z}$. Rotations over the three axes are executed online during training, so angles α, β, γ are randomly generated for each sample. We show the effects of this augmentation with various angles later in experimentation.

E. Data Collection

Data was collected executing grasp trials, as shown in Fig. 4. GeoGrasp [42], [43] was used to compute grasping points on the 3D point clouds. This method finds grasps on unknown objects using a 3D point cloud with a partial view, which fits our setup. GeoGrasp tends to find grasps around the cloud's centroid. In order to avoid fitting our learning to this type of grasps only, we approximated the object's main axis using Principal Component Analysis (PCA) and then randomly moved the grasps along this axis. This resulted in grasps being performed all over the objects' surface (Fig. 5)

We saved from each grasp trial: the 3D point cloud of the object $\mathbb{C}$, the grasp configuration Θ_F and the tactile readings at the moment of contact, so a sample is a tuple $\mathbb{S} = \langle \mathbb{C}, \Theta_F, PDC_{mf}, PDC_{th}, E_{mf}, E_{th} \rangle$. The robotic fingers were closed so they would come into contact with the object on the computed grasping points without exceeding the torque limits of the joints. Note that this does not mean that grasps were executed with a constant force for all the items. We recorded grasps with different tactile responses (Fig. 6) but the torque limits were our upper bound in order to avoid breaking the robot. We followed these steps to collect a sample:

1) Move the robot away from the table to allow the camera to get a clean view of the scene.
2) Place a single object on the table with a random orientation, as shown in Fig. 5.
3) Capture a raw point cloud $\mathbb{C}$ using the depth camera.
4) Send it to the modified GeoGrasp in order to obtain a grasp configuration $\Theta_F = \{g_1, g_2, R\}$.
5) Save the segmented cloud $\mathbb{C}$ and the grasp data Θ_F.

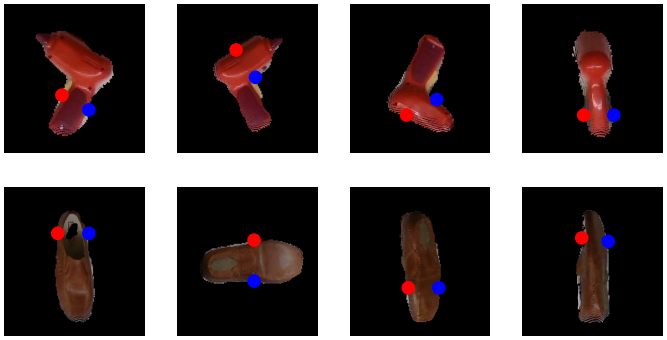

Fig. 5. Example of various poses seen from the camera and grasps for two of the objects in the set during the collection of data for this work.

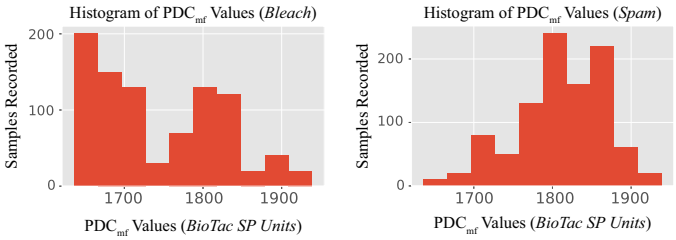

Fig. 6. Histogram of PDC_{mf} values for two items in our dataset showing that different pressure values have been recorded.

6) Calculate the Inverse Kinematics (IK) and plan a trajectory so that the robot reaches the target grasp.
7) Execute the trajectory.
8) After making contact, save the general pressure PDC and the electrodes values E from both fingers.
9) Release the grasp and repeat from step 1.

Grasps were executed on a collection of 16 objects (Fig. 7). These objects were selected because they represent various geometrical shapes and also different degrees of stiffness. For our training set, we used 8 items from the YCB item set [44], plus 4 soft items that we added because YCB objects are mostly rigid. Soft items were included to increase the variability in the sensed tactile responses, since these items deform under contact. We specifically recorded 1000 samples for each of the following objects: 4 cylinder-like objects (can of Pringles, can of coffee, bottle of bleach, can of Campbell's soup), 4 box-like objects (box of Cheezits, box of sugar, can of Spam, piece of wood) and 4 soft objects (stuffed Minion, flat ball, sponge, stuffed volley ball). Therefore, our training set is

Fig. 7. Collection of 16 objects used in our experiments.

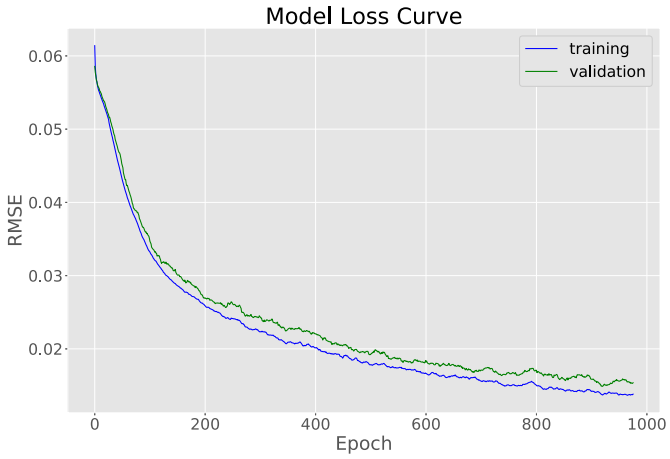

Fig. 8. Evolution of the average loss rate (RMSE) as regard the generation of PDC_{th} values with 5-fold cross-validation.

not biased towards neither a single geometrical primitive nor rigid objects only. In addition, we recorded a set containing 500 samples for each of the following 4 novel items: plastic toy train, plastic toy drill, stuffed rugby ball and leather shoe. These new objects were selected because they have novel shapes (e.g. plastic toy drill from YCB set) and degrees of stiffness (e.g. leather shoe). In total, our sets contain 12000 and 2000 samples respectively[1].

V. Experiments

The proposed learning network was trained using Root Mean Square Error (RMSE) as loss function, where y_t is the target tactile value, $\hat{y}_t$ is the generated value and n is the number of samples in the batch:

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2} \quad (1)$$

We split the training samples (12000) into two balanced sets: 75% were used for training and the remaining 25% for the test set, ensuring that each of the 12 objects were equally represented within the sets. Experiments on the 9000 training samples were carried out using 5-fold cross validation in order to extract better error estimates and to avoid overfitting [45]. Batch normalisation was also applied to improve performance. We used the Adam optimiser with a learning rate equal to 0.01 and the batch size was equal to 10, owing to our dataset size and our hardware. Finally, the experiments were run on a PC with an Intel i7-8700K CPU at 3.7GHz, 32 GiB DDR4 RAM and two GeForce GTX 1080Ti GPU, and running Ubuntu 16.04, Python 3.6.9, CUDA 10.0 and PyTorch 1.2.0.

We worked with the PDC_{th} data in order to find the best configuration of architecture hyperparameters (Section V-A), input representation (Section V-B), normalisation and augmentation methods (Section V-C). Since the BioTac SP sensors have slightly different behaviour and data ranges, we trained one network for each sensor and tactile modality for our final test experiments (Sections V-D and V-E). This means that one network is trained with PDC_{th} values, another with PDC_{mf}, another with E_{th} and a fourth one with E_{mf}. However, we used the same architecture and hyperparameters for all of them, based on the results obtained from PDC_{th}.

[1]Data available at: https://github.com/yayaneath/vision2tactile

TABLE I
Average 5-fold cross-validation RMSE as regards the generation of PDC_{th} values for our architectures. We omit the final layer that produces the regressed output.

Architecture	Fully-Connected ReLUs	Loss (RMSE $\pm$ STD)
FC1	[1024]	0.02598 $\pm$ 0.00043
FC2	[1024, 512]	0.02586 $\pm$ 0.00084
FC3	[1024, 512, 256]	**0.02570 $\pm$ 0.00076**
FC4	[1024, 512, 256, 128]	0.02744 $\pm$ 0.00069
FC5	[1024, 512, 256, 128, 64]	0.02652 $\pm$ 0.00092
FC6	[1024, 512, 256, 128, 64, 32]	0.02660 $\pm$ 0.00062

A. Architecture and Hyperparameters

We first trained a basic architecture for tuning the number of epochs. Results are shown in Fig. 8. This was obtained by training a network with a PointNet feature calculator and three FC layers with ReLUs ($FC3$ in Table I). The input were point clouds with only 3D coordinates, downsampled to 800 points, and no grasp data. We used the PDC_{th} data from the 9000 training samples and ran the training loop for 1000 epochs using 5-fold cross-validation. As can be seen, loss began to converge at 500 epochs. However, the change in loss between 250 epochs and 500 epochs is less than 0.01 points. We, therefore, carry out the remaining of our experiments using 250 epochs in order to avoid overfitting.

We then experimented with the FC layers after the PointNet feature calculator. Six architectures were tested with varying depths and sizes, as detailed in Table I. These results were obtained by training each architecture on the 9000 PDC_{th} samples with 5-fold cross-validation for 250 epochs. The shallowest network $FC1$ underfits the problem. As we increase the complexity of the architecture, the error decreases. However, from $FC4$ and above, the error increases again. As a result, we use $FC3$ as our base architecture in the remaining experiments. This architecture is shown in Fig. 3.

TABLE II
AVERAGE 5-FOLD CROSS-VALIDATION RMSE AS REGARDS THE GENERATION OF PDC_{th} VALUES BY CLOUD SIZE. TIME REQUIRED TO PROCESS A SINGLE EPOCH IS ALSO SHOWN.

Cloud Size	Loss (RMSE $\pm$ STD)	Time/epoch (s)
250	0.03010 $\pm$ 0.00105	8
500	0.02790 $\pm$ 0.00115	14
750	**0.02717 $\pm$ 0.00134**	21

B. Visual and Grasp Representations

We first experimented with the number of points in the downsampled clouds. This is a parameter of the PointNet feature calculator that must be fixed and equal to every cloud. Since the smallest cloud in our dataset held 814 points, we limited ourselves to a maximum of 750 points. Three different sizes were tested: $250, 500$ and 750. Table II shows the results obtained for 9000 PDC_{th} samples. As the number of points used is reduced, the loss increases, which is a consequence of losing more information during the sampling. Hence, the best performance should be obtained from larger clouds. However, increasing their size is also detrimental to the execution time: a single epoch using clouds with 750 points takes 162.5% more time than if the clouds have 250 points. In the remaining experiments, we downsample the clouds to 750 points to maintain as much information as possible. We do not use larger sizes because that would require inventing points for smaller clouds and it would also significantly increase execution times.

Six combinations of visual and grasp representations were tested as input for our system:

1) *XYZ*, which is a cloud in which points contain only 3D coordinates, so a single sample has the shape 750×3.
2) *RGB*, which extends *XYZ* by adding RGB colour channels, meaning that a single sample has the shape 750×6.
3) *XYZCont*, which extends *XYZ* with grasp contact data (points g_1 and g_2), so that a single sample contains a cloud with shape 750×3 and a vector with shape 1×6 that concatenates the 3D coordinates of the two contact points.
4) *RGBCont*, which extends *RGB* with grasp contact data, meaning that a single sample contains a cloud with shape 750×6 and a vector with shape 1×6 that concatenates the 3D coordinates of the two contact points.
5) *XYZPose*, which extends *XYZCont* with pose information (rotation matrix R of the wrist), so that a single sample contains a cloud with shape 750×3 and a vector with shape 1×15 that concatenates the 3D coordinates of the two contact points and the three axis defining R.
6) *RGBPose*, which extends *RGBCont* with pose information, such that a single sample contains a cloud with shape 750×6 and a vector with shape 1×15 that concatenates the 3D coordinates of the two contact points and the three axis defining R.

We used the 9000 PDC_{th} samples to train the $FC3$ architecture for 250 epochs, using 5-fold cross-validation on these inputs. The results are shown in Fig. 9. We expected that the addition of colour data to *XYZ* would always result in lower errors, but the *RGB* configuration yielded higher values. This could have been owing to an increased complexity in the input for our task: since our items have different coloured textures over the entirety of their surfaces, the use of *RGB* representation increases the variability in the input. In comparison, learning from a simpler input such as *XYZ* allows the learner to fit the task further. A similar result can be found when checking *XYZCont* and *RGBCont*. This suggests that adding colour information does not guarantee an improved performance.

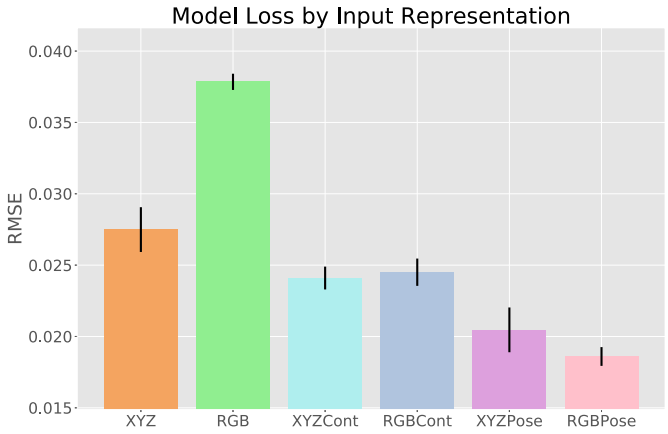

Fig. 9. Average 5-fold cross-validation RMSE as regards the generation of PDC_{th} values using different input representations.

TABLE III
AVERAGE 5-FOLD CROSS-VALIDATION RMSE AS REGARDS THE GENERATION OF PDC_{th} VALUES USING NORMALISATION METHODS.

Normalisation	Loss (RMSE $\pm$ STD)
None	0.01859 $\pm$ 0.00065
Unit Sphere	0.01183 $\pm$ 0.00073
Min-Max Scaling	**0.01141 $\pm$ 0.00076**
Wrist Frame	0.02260 $\pm$ 0.00100

In contrast, *XYZCont* and *RGBCont* provided lower error rates than *XYZ* and *RGB*. Hence, adding contact information improves the performance of the tactile generator. The best results were obtained by adding the rotation of the wrist. This confirms that adding grasp-related information improves performance in the case of our tactile data generation task. Consequently, we run remaining experiments using *RGBPose*.

C. Normalisation and Augmentation

Table III lists the results obtained from experiments carried out to compare normalisation methods. As can be seen, the use of a normalisation method always lowered the error, with the exception of the *Wrist Frame* technique. It could have performed less well because this method transforms points to a reference frame, but it does not really normalise their value ranges. The best results were obtained using *Min-Max scaling*, which reduced error to almost 22%. Thereby, following experiments use this normalisation.

We experimented with four ranges of angles for testing data augmentation of the input data with rotations. That is, we trained one network randomly rotating the Cartesian coordinates of the input between $-45°$ and $45°$, another was

TABLE IV
AVERAGE 5-FOLD CROSS-VALIDATION RMSE AS REGARDS THE GENERATION OF PDC_{th} VALUES FOR EACH AUGMENTATION METHOD. ROTATION WITH $\pm 0°$ MEANS NO AUGMENTATION.

Rotation Angle	Loss (RMSE $\pm$ STD))
$0°$	**0.01141 $\pm$ 0.00076**
$[-45°, 45°]$	0.02026 $\pm$ 0.00073
$[-90°, 90°]$	0.02545 $\pm$ 0.00094
$[-135°, 135°]$	0.02788 $\pm$ 0.00109
$[-180°, 180°]$	0.02671 $\pm$ 0.00073

trained with rotations in the range $[-90°, 90°]$, a third one was trained with the range $[-135°, 135°]$ and the last was trained with the range $[-180°, 180°]$. In contrast to our expectations, increasing the variability of the input using this method was counter-productive, since validation error increased by more than 77%. This could be the result of increasing the complexity of the input and introducing noise, thus misleading the network as regards learning a correct representation for our task.

D. Test Set

We discovered from previous experiments that best results were obtained by training a $FC3$ network with downsampled clouds of 750 points for 250 epochs using the *RGBPose* input representation, normalised using *Min-Max Scaling*. In this experiment, we trained this system again but we evaluated it with the remaining 3000 samples for each tactile modality: $PDC_{th}, PDC_{mf}, E_{th}, E_{mf}$. We show in Fig. 10 the average results obtained from 5 independent iterations of training with 9000 samples and then testing with these 3000 testing samples.

We obtained an average RMSE equal to 0.05496 upon generating PDC_{th} values. This value, when scaled back to the discrete range of this modality, equals 49 units in its custom values. In context, this can be considered a low error, since values for this modality are in the discrete range $[2500, 3400]$. As for PDC_{mf}, the error obtained (0.06170) equals 43 units scaled back to the range of this modality ($[1600, 2300]$), which can also be considered low in this context. As can be seen, our proposal produces lower error PDC_{th} than PDC_{mf} responses. Learning to generate PDC_{th} might be easier owing to the orientation of the thumb on the Shadow Hand: it is slightly turned in, like a human thumb, meaning that our gripper-like grasps could not completely touch items with this finger. Instead, the thumb mostly contacted them with one side of its surface. This might have translated into similar general forces experienced by the thumb sensor, thus reducing the complexity of learning from these data.

With regard to E_{th} responses, we obtained an average RMSE equal to 0.06099, which are 213 units in the discrete range of values of this modality ($[100, 3600]$). In the case of E_{mf}, we obtained an average RMSE equal to 0.05922, or 195 in its modality range ($[500, 3800]$). Our proposal yielded lower error rates for the middle finger than for the thumb. This could be related to the orientation of the fingers mentioned above. The middle finger made contact with the objects with its whole surface. This might have resulted in unique activations of the electrodes: that is, the E_{mf} responses were more characteristic

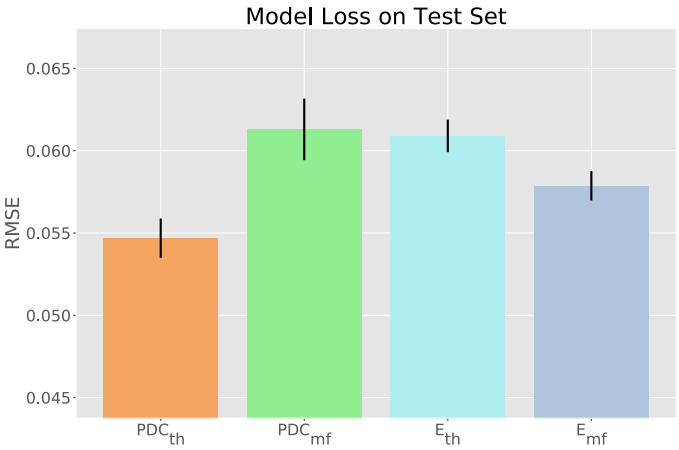

Fig. 10. Average RMSE from 5 runs of training and regressing each tactile modality using the test set (3000 samples of known objects). For PDC, this is the average error of 1 pressure value regressed on each run. For E, this is the average error of 24 electrodes values regressed on each run.

depending on the grasp and the item. If you touch the tip of a fork with your fingertip, you will probably distinguish the tines and the spaces between them. However, if you touch the fork with just a side of your fingertip, it might be more difficult to distinguish it from the tip of a spoon if all you feel is a solid surface. This is the way in which the thumb made contact with items, so its electrodes could not encode much information depending on the object and the grasp. As a result, it was more complex learning from E_{th} than E_{mf} data because its data were less discriminative.

We conclude that learning to generate tactile data is a task with different degrees of complexity depending on the modality used and the way in which the BioTac SP makes contact with target objects. On the one hand, if the sensor contacts the manipulated object with its whole surface, then its electrodes seem to register a discriminative signature of it such that our system learns to generate low-error responses. However, the general pressure recorded is less specific and, therefore, more difficult to generate given visual and grasp data. On the other hand, if the sensor barely comes into contact with part of its surface, then the electrodes cannot encode a discriminative signature. Consequently, producing general pressure values becomes less complex.

E. Novel Objects

We further verified the generalisation capabilities of the proposed system to new items by training our best configuration on the whole training set (12000 samples) and evaluating it with the set of 4 novel items (2000 samples). We show in Fig. 11 the average RMSE obtained from 5 independent iterations of training and then testing on this set. We ran various iterations in order to extract conclusions from various trials rather than a single lucky run.

The average error for PDC_{th} increased from 0.05496 (49 units) with the test set to 0.08289 (74 units) with the set of novel objects, which is an increase of 51%. The PDC_{mf} error increased by 73% from 0.06170 (43 units) to 0.10650 (74 units) with this new set. As for E_{th}, it increased by 43%

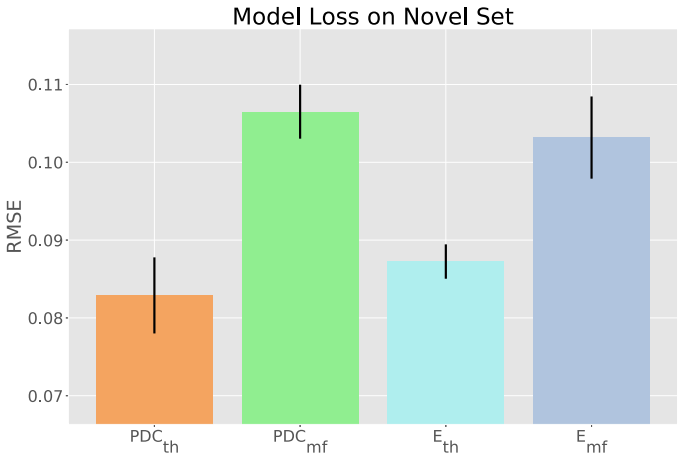

Fig. 11. Average RMSE obtained after 5 runs of training and regressing each tactile modality using the novel set (2000 samples of novel objects). For PDC, this is the average error of 1 pressure value regressed on each run. For E, this is the average error of 24 electrodes values regressed on each run.

from 0.06099 (213 units) to 0.08724 (305 units). Finally, E_{mf} increased by 74% from 0.05922 (195 units) to 0.10317 (340 units). These results demonstrate that our system had more difficulty processing the novel items, since the loss of each data modality increased by at least 43%, peaking at 74%.

Increased error rates were expected, since generalisation to novel samples is a challenge in deep neural networks. Moreover, the objects selected for our novel set were different as regards their shape (e.g. plastic toy drill) and materials (e.g. leather shoe). It is, however, possible to identify a trend: the error rates increased by similar percentages for the data modalities concerning the same finger. The increase in error as regards generating PDC and E values for the thumb are 51% and 43%, respectively. These increases are 73% and 74% for the same modalities for the middle finger. Producing readings for this finger might have been more difficult because newer tactile patterns might have been recorded with these novel items. In contrast, since the thumb continued to contact items with the same side of its fingertip, these novel items did not produce completely different tactile patterns.

In addition, the differences in the loss between data modalities for the same finger had a similar trend as that of the test set. That is, producing E values for the thumb provided higher error rates than producing PDC for that same finger. This can be verified in the results obtained for the test set (Fig. 10) and the novel set (Fig. 11). In the case of the middle finger, generating E values provided lower error rates than generating PDC values. This confirms that generating tactile data with the BioTac SP is a task with different degrees of complexity. This depends on the modality of data used and the way in which the sensor establishes contact with objects. According to our results, when the sensor makes contact with the whole of its surface, discriminative patterns are recorded by the electrodes, which lowers the complexity of producing E values when compared to PDC. However, if the sensor contacts objects with one side of its surface, these discriminative patterns are not that clear in the electrodes, and generating PDC becomes less complex in comparison.

VI. Limitations

The major limitation of our work is the unstable behaviour of the BioTac SP. These sensors are highly responsive to a wide range of contacts and forces. However, two sensors do not provide exactly the same numeric values, nor do they behave identically. There are slight differences as regards in the amount of liquid inside of them owing to their construction, and this has a high impact on the responses provided. In addition, the ranges of values are different from sensor to sensor. As a result, tactile patterns detected on one sensor could not be found on other sensors, thus limiting the transferability of our models.

A second limitation of our proposal is the ability to visually capture tactile clues. Using 3D point clouds as the only description of the item might not be sufficient to predict accurate tactile responses. For example, a can made of steel and a can made of cardboard would produce different contact responses. However, their geometries are the same and they may have similar colours, if painted. We are, therefore, limited as regards the amount of information processed about the object. Nevertheless, if the item is previously recognised, we could include in the learning pipeline a vector of characteristics describing it, as Abderrahmane *et al.* [32] showed in their work. Then, our system would receive more information about the object.

Finally, another factor affecting the difficulty of this task is the robotic hand used and, therefore, the grasp configuration. In our case, we used the BioTac SP sensors on the middle finger and thumb of a Shadow Dexterous hand. The tactile responses generated from these fingers were different because they were oriented differently with respect to the contacted objects. As a result, we have not approached one problem, but four instead: the generation of global pressure values and electrodes responses for the middle finger and for the thumb separately. We have shown that each data modality on each finger configures a task with a different degree of complexity.

VII. Conclusion

This work presents one of the first approaches for the generation of tactile responses using 3D visual perception and grasp configuration data. We propose to learn to regress tactile data from two BioTac SP tactile sensors using a neural network based on PointNet. Our system uses 3D point clouds, including Cartesian coordinates and colour channels, and data from two-fingered grasps in order to model the behaviour of our tactile sensors. Two sources of data are considered: the global pressure value PDC, which is a single number; and electrodes E, which are 24 values that are related owing to their vicinity inside the sensor. Therefore, we cover a regression task.

In experimentation, we used 12 objects (12000 samples) to find the best architecture for our task and 4 novel items (2000 samples) to verify the generalisation capabilities of our proposal with new objects. Our experiments have proved that it is possible to regress tactile responses by combining 3D point clouds with grasp-related data like contact points and the pose of the tool. As a result, our system generates responses with low error values when compared to the ranges of values of

each modality. This tactile generator could be applied to find grasp candidates on seen objects or to enrich the input of an object detector, thus providing an extra perception modality.

We conclude that generating tactile data with the BioTac SP is a task with different degrees of complexity. This complexity depends on the modality of the data used and the way in which the sensor comes into contact with objects. According to our results, when the entire surface of the sensor makes contact, then discriminative patterns are recorded by the electrodes, which lowers the complexity of producing E values when compared to PDC. However, if only a side of its surface comes into contact with objects, these discriminative patterns are not that clear in the electrodes, so generating PDC becomes less complex in comparison. Our work is consequently limited by the way in which the Shadow Hand configures its fingers in order to make contact with items.

In future work, we would like to record more data on new items, since we are using deep learning networks which are known for being data hungry models. We also plan to explore ways to exploit unlabelled data: 3D point clouds and grasp configurations with no target tactile responses, which are easy and fast to collect. Moreover, this work paves the way towards developing a grasp generator that will improve the stability of the candidate grasps by using this system as a means to measure their quality. Finally, we plan to experiment with synthetic data in order to verify whether the models learnt for the sensor can be transferred from a real system to a simulated environment, or whether our data can be increased by using this synthetic information.

REFERENCES

[1] R. W. Fleming, "Visual perception of materials and their properties," *Vision Research*, vol. 94, pp. 62–75, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.visres.2013.11.004

[2] Z. Kappassov, J.-A. Corrales, and V. Perdereau, "Tactile sensing in dexterous robot hands — Review," *Robotics and Autonomous Systems*, vol. 74, pp. 195–220, dec 2015. [Online]. Available: http://dx.doi.org/10.1016/j.robot.2015.07.015 http://linkinghub.elsevier.com/retrieve/pii/S0921889015001621

[3] N. F. Lepora, A. Church, C. de Kerckhove, R. Hadsell, and J. Lloyd, "From Pixels to Percepts: Highly Robust Edge Perception and Contour Following Using Deep Learning and an Optical Biomimetic Tactile Sensor," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2101–2107, apr 2019. [Online]. Available: http://arxiv.org/abs/1812.02941 https://ieeexplore.ieee.org/document/8641397/

[4] V. Prado da Fonseca, T. E. Alves de Oliveira, and E. M. Petriu, "Estimating the Orientation of Objects from Tactile Sensing Data Using Machine Learning Methods and Visual Frames of Reference," *Sensors*, vol. 19, no. 10, pp. 1–20, 2019.

[5] X. Li, K. Zhao, C. Lu, and Y. Wang, "Quantitative motion detection of in-hand objects for robotic grasp manipulation," *International Journal of Advanced Robotic Systems*, vol. 16, no. 3, pp. 1–12, 2019.

[6] M. Kaboli, K. Yao, D. Feng, and G. Cheng, "Tactile-based active object discrimination and target object search in an unknown workspace," *Autonomous Robots*, no. January, pp. 1–30, 2018.

[7] M. Bauza, O. Canal, and A. Rodriguez, "Tactile Mapping and Localization from High-Resolution Tactile Imprints," in *2019 IEEE International Conference on Robotics and Automation (ICRA)*, 2019. [Online]. Available: http://arxiv.org/abs/1904.10944

[8] S. Luo, J. Bimbo, R. Dahiya, and H. Liu, "Robotic tactile perception of object properties: A review," *Mechatronics*, vol. 48, no. May, pp. 54–67, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.mechatronics.2017.11.002

[9] H. Liu, Y. Yu, F. Sun, and J. Gu, "Visual–Tactile Fusion for Object Recognition," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 2, pp. 996–1008, apr 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7462208/

[10] S. Luo, W. Yuan, E. Adelson, A. G. Cohn, and R. Fuentes, "ViTac: Feature Sharing Between Vision and Tactile Sensing for Cloth Texture Recognition," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, may 2018, pp. 2722–2727. [Online]. Available: https://ieeexplore.ieee.org/document/8460494/

[11] J. Lin, R. Calandra, and S. Levine, "Learning to Identify Object Instances by Touch: Tactile Recognition via Multimodal Matching," in *2019 IEEE International Conference on Robotics and Automation (ICRA)*, mar 2019. [Online]. Available: http://arxiv.org/abs/1903.03591

[12] Z. Abderrahmane, G. Ganesh, A. Crosnier, and A. Cherubini, "Haptic Zero-Shot Learning: Recognition of objects never touched before," *Robotics and Autonomous Systems*, vol. 105, pp. 11–25, 2018. [Online]. Available: https://doi.org/10.1016/j.robot.2018.03.002

[13] E. Velasco, B. S. Zapata-Impata, P. Gil, and F. Torres, "Clasificación de objetos usando percepción bimodal de palpación única en acciones de agarre robótico," *Revista Iberoamericana de Automatica e Informatica Industrial*, no. April, pp. 1–12, 2019. [Online]. Available: https://polipapers.upv.es/index.php/RIAI/article/view/10923

[14] S. Luo, W. Mou, K. Althoefer, and H. Liu, "iCLAP: shape recognition by combining proprioception and touch sensing," *Autonomous Robots*, vol. 43, no. 4, pp. 993–1004, 2019.

[15] S. Dong, D. Ma, E. Donlon, and A. Rodriguez, "Maintaining Grasps within Slipping Bound by Monitoring Incipient Slip," in *2019 IEEE International Conference on Robotics and Automation (ICRA)*, 2019. [Online]. Available: http://arxiv.org/abs/1810.13381

[16] B. S. Zapata-Impata, P. Gil, and F. Torres, "Non-Matrix Tactile Sensors: How Can Be Exploited Their Local Connectivity For Predicting Grasp Stability?" in *IEEE/RSJ IROS 2018 Workshop RoboTac: New Progress in Tactile Perception and Learning in Robotics*, 2018, pp. 1–4. [Online]. Available: http://arxiv.org/abs/1809.05551

[17] M. A. Abd, I. J. Gonzalez, T. C. Colestock, B. A. Kent, and E. D. Engeberg, "Direction of slip detection for adaptive grasp force control with a dexterous robotic hand," *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*, vol. 2018-July, pp. 21–27, 2018.

[18] B. Zapata-Impata, P. Gil, and F. Torres, "Learning Spatio Temporal Tactile Features with a ConvLSTM for the Direction Of Slip Detection," *Sensors*, vol. 19, no. 3, p. 523, jan 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/3/523 http://www.mdpi.com/1424-8220/19/3/523

[19] V. Arapi, Y. Zhang, G. Averta, M. Catalano, D. Rus, C. Della Santina, and M. Bianchi, "To grasp or not to grasp: an end-to-end deep-learning approach for predicting grasping failures in soft hands," in *Proceedings of the IEEE International Conference on Soft Robotics (RoboSoft)*, 04 2020.

[20] J. Kwiatkowski, D. Cockburn, and V. Duchaine, "Grasp stability assessment through the fusion of proprioception and tactile signals using convolutional neural networks," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, sep 2017, pp. 286–292. [Online]. Available: http://ieeexplore.ieee.org/document/8202170/

[21] R. Calandra, A. Owens, M. Upadhyaya, W. Yuan, J. Lin, E. H. Adelson, and S. Levine, "The Feeling of Success: Does Touch Sensing Help Predict Grasp Outcomes?" in *Proceedings of the 1st Annual Conference on Robot Learning*, vol. 78, 2017, pp. 314—-323. [Online]. Available: http://proceedings.mlr.press/v78/calandra17a.html

[22] J. Li, S. Dong, and E. Adelson, "Slip Detection with Combined Tactile and Visual Information," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, may 2018, pp. 7772–7777. [Online]. Available: http://arxiv.org/abs/1802.10153 https://ieeexplore.ieee.org/document/8460495/

[23] R. Calandra, A. Owens, D. Jayaraman, J. Lin, W. Yuan, J. Malik, E. H. Adelson, and S. Levine, "More Than a Feeling: Learning to Grasp and Regrasp Using Vision and Touch," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3300–3307, oct 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8403291/

[24] S. H. Huang, M. Zambelli, J. Kay, M. F. Martins, Y. Tassa, P. M. Pilarski, and R. Hadsell, "Learning Gentle Object Manipulation with Curiosity-Driven Deep Reinforcement Learning," *arXiv*, 2019. [Online]. Available: http://arxiv.org/abs/1903.08542

[25] G. Sutanto, N. Ratliff, B. Sundaralingam, Y. Chebotar, Z. Su, A. Handa, and D. Fox, "Learning Latent Space Dynamics for Tactile Servoing," in *2019 IEEE International Conference on Robotics and Automation (ICRA)*, 2019. [Online]. Available: http://arxiv.org/abs/1811.03704

[26] S. Tian, F. Ebert, D. Jayaraman, M. Mudigonda, C. Finn, R. Calandra, and S. Levine, "Manipulation by Feel: Touch-Based Control with Deep Predictive Models," in *2019 IEEE International Conference*

on Robotics and Automation (ICRA), mar 2019. [Online]. Available: http://arxiv.org/abs/1903.04128

[27] A. Montaño and R. Suárez, "Manipulation of unknown objects to improve the grasp quality using tactile information," *Sensors*, vol. 18, no. 5, 2018.

[28] F. Veiga, J. Peters, and T. Hermans, "Grip Stabilization of Novel Objects using Slip Prediction," *IEEE Transactions on Haptics*, vol. 11, no. 4, pp. 531–542, 2018.

[29] T. H. Pham, A. Kheddar, A. Qammaz, and A. A. Argyros, "Towards force sensing from vision: Observing hand-object interactions to infer manipulation forces," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June, pp. 2810–2819, 2015.

[30] T. H. Pham, N. Kyriazis, A. A. Argyros, and A. Kheddar, "Hand-Object Contact Force Estimation from Markerless Visual Tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 2883–2896, 2018.

[31] H. Shin, H. Cho, D. Kim, D.-k. Ko, S.-C. Lim, and W. Hwang., "Sequential Image-based Attention Network for Inferring Force Estimation without Haptic Sensor," *IEEE Access*, vol. 7, pp. 1–1, 2019.

[32] Z. Abderrahmane, G. Ganesh, A. Crosnier, and A. Cherubini, "A Deep Learning Framework for Tactile Recognition of Known as well as Novel Objects," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2019.

[33] F. R. Hogan, M. Bauza, O. Canal, E. Donlon, and A. Rodriguez, "Tactile Regrasp: Grasp Adjustments via Simulated Tactile Transformations," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Madrid, Spain: IEEE, oct 2018, pp. 2963–2970. [Online]. Available: http://arxiv.org/abs/1803.01940 https://ieeexplore.ieee.org/document/8593528/

[34] J.-T. Lee, D. Bollegala, and S. Luo, ""Touching to See" and "Seeing to Feel": Robotic Cross-modal Sensory Data Generation for Visual-Tactile Perception," in *2019 IEEE International Conference on Robotics and Automation (ICRA)*, feb 2019. [Online]. Available: http://arxiv.org/abs/1902.06273

[35] Y. Li, J.-Y. Zhu, R. Tedrake, and A. Torralba, "Connecting Touch and Vision via Cross-Modal Prediction," in *2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. [Online]. Available: http://arxiv.org/abs/1906.06322

[36] Syntouch, "BioTac SP," 2018. [Online]. Available: https://www.syntouchinc.com/en/sensor-technology/

[37] Shadow Robot Company, "Shadow Dexterous Hand," 2018. [Online]. Available: http://www.shadowrobot.com/products/dexterous-hand/

[38] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," in *ICRA workshop on open source software*, vol. 3, no. 3.2, 2009, p. 5.

[39] I. A. Sucan and S. Chitta, "Moveit!" *Online at http://moveit.ros.org*, 2020.

[40] B. Sundaralingam, A. S. Lambert, A. Handa, B. Boots, T. Hermans, S. Birchfield, N. Ratliff, and D. Fox, "Robust Learning of Tactile Force Estimation through Robot Interaction," in *2019 IEEE International Conference on Robotics and Automation (ICRA)*, 2019. [Online]. Available: http://arxiv.org/abs/1810.06187

[41] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2017-Janua. IEEE, jul 2017, pp. 77–85. [Online]. Available: http://ieeexplore.ieee.org/document/8099499/

[42] B. S. Zapata-Impata, C. M. Mateo, P. Gil, and J. Pomares, "Using Geometry to Detect Grasping Points on 3D Unknown Point Cloud," in *Proceedings of the 14th International Conference on Informatics in Control, Automation and Robotics (ICINCO) 2017*, vol. 2. SCITEPRESS - Science and Technology Publications, 2017, pp. 154–161. [Online]. Available: doi=10.5220/0006470701540161

[43] B. S. Zapata-Impata, P. Gil, J. Pomares, and F. Torres, "Fast geometry-based computation of grasping points on three-dimensional point clouds," *International Journal of Advanced Robotic Systems*, vol. 16, no. 1, jan 2019. [Online]. Available: http://journals.sagepub.com/doi/10.1177/1729881419831846

[44] B. Calli, A. Singh, J. Bruce, A. Walsman, K. Konolige, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Yale-CMU-Berkeley dataset for robotic manipulation research," *The International Journal of Robotics Research*, vol. 36, no. 3, pp. 261–268, 2017. [Online]. Available: http://journals.sagepub.com/doi/10.1177/0278364917700714

[45] A. Blum, A. Kalai, and J. Langford, "Beating the hold-out: bounds for K-fold and progressive cross-validation," *Proceedings of the Annual ACM Conference on Computational Learning Theory*, pp. 203–208, 1999.

Brayan S. Zapata-Impata (M'17) received the B.Sc. and M.Sc. degree in Computer Engineering from the University of Alicante, Spain, in 2015 and 2017, respectively. He has been working on his Ph.D. degree since 2016 and currently is a researcher of the Department of Physics, System Engineering and Signal Theory of the University of Alicante. His research is focused on robotic grasping and manipulation, based on visual and tactile perception and using machine learning techniques in order to provide robots with intelligent skills. His work is currently supported by the Spanish Ministry of Science, Innovation and Universities through a national-wide, competitive scholarship. He is Vice President of the Spanish Chapter of IEEE Young Professionals and he is also member of the IEEE Robotics and Automation Society.

Pablo Gil (M'12-SM'14) received a B.Sc. degree in Computer Science Engineering and a Ph.D. degree from the University of Alicante, Spain, in 1999 and 2008, respectively. He holds a tenured position as Associate Professor at University of Alicante. From 2016 to 2018, he was the secretary of Computer Science Research Institut and he is currently the head from 2018. He has also been a Researcher on over 18 research and development projects funded by the European Commission, Spanish Government agencies and private companies. His research interests include computer vision, 3-D vision, deep learning and perception for robots. He has co-authored more than 100 works on those topics. Dr. Gil is a member of the Spanish Automatic Committee of IFAC and a senior member of the IEEE Robotics and Automation Society, Education Society and the IEEE Sensor Council. He is the secretary of IEEE-RAS Spanish Chapter from 2018.

Youcef Mezouar received the Ph.D. degrees in automation and computer science from the University of Rennes 1, France in 2001. He obtained the Habilitation Degree (HDR - Habilitation à Diriger des Recherches) from Université Blaise Pascal, Clermont-Ferrand, France, in November 2009. He spent one year as Postdoctoral Associate in the Robotic Lab of the Computer Science Department of Columbia University, New York. He was assistant professor from 2002 to 2011 in the Physics Department of Blaise Pascal University, Clermont-Ferrand, France. He holds a tenured position as Full Professor at SIGMA'Clermont (IFMA until 2016) since 2012. At SIGMA, he is the head of the Machines, Mechanisms and Systems Department. He is performing research at the MACCS (Modeling, Autonomy and Control of ComplEx System) team of the Image, Perception System and Robotics (IPSR) of Institut Pascal. He currently co-leads the IPSR reseach group (over 80 persons) and the MACCS team (around 20 persons). His research interests include sensor-based control with applications to dexterous manipulation and mobile robots coordination. He co-authored more than 160 papers on the topics.

Fernando Torres (M'02-SM'12) received a B.Sc. degree in Industrial Engineering and a Ph.D. degree from the Polytechnic University of Madrid, Spain, in 1991 and 1995, respectively. He holds tenured position as Full Professor at University of Alicante, where he is the head of AUROVA (Automatics, Robotics and Artificial Vision) research group. His research focuses on automation and robotics (intelligent robotic manipulation, visual control, perception systems, neuro-robotics, field robots, advanced automation for industry 4.0). In these lines, he currently has more than 50 publications in JCR-ISI journals and more than 100 papers in international conferences. In addition, he is a member of TC 5.1 and TC 9.4 of the IFAC, a Senior Member of the IEEE and a member of CEA. In the past, he was deputy director of the Department of Physics, Systems Engineering and Signal Theory at University of Alicante. From 2009 to 2011, he has been deputy coordinator of the area of Electrical, Electronic and Automatic Engineering (IEL) of the National Agency of Evaluation and Prospective (ANEP) and Coordinator from 2012 to February 2016. Since July 2018, he is the current Coordinator of the area of Electrical, Electronic and Automatic (IEA) of the Spanish Agency of Statal Research (AEI).