2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Prediction of Tactile Perception from Vision on Deformable Objects

Brayan S. Zapata-Impata[1] and Pablo Gil[1]

Abstract— **Through the use of tactile perception, a manipulator can estimate the stability of its grip, among others. However, tactile sensors are only activated upon contact. In contrast, humans can estimate the feeling of touching an object from its visual appearance. Providing robots with this ability to generate tactile perception from vision is desirable to achieve autonomy. To accomplish this, we propose using a Generative Adversarial Network. Our system learns to generate tactile responses using as stimulus a visual representation of the object and target grasping data. Since collecting labeled samples of robotic tactile responses consumes hardware resources and time, we apply semi-supervised techniques. For this work, we collected 4000 samples with 4 deformable items and experiment with 4 tactile modalities.**

I. INTRODUCTION

Thanks to the sense of touch, humans can find an object inside a drawer or keep a stable grip on a glass while walking. In combination with vision, humans can estimate physical attributes of objects without touching them. It is argued that the human brain builds statistical generative models that capture visual clues, which are exploited in order to predict such properties [1]. Providing robots with this ability to generate tactile perception from vision is desirable to achieve autonomy. Thus, a robot could recognize objects not only from their visual appearance but also using predicted tactile responses. Remarkably, this would be accomplished without making contact with the object.

Our main goal is to provide a system that regresses robotic tactile responses from vision, so multi-modal methods can be used before moving the robot. This system could even provide robots with the sense of touch in the case they do not have tactile sensors installed. However, gathering labeled samples of tactile responses with a robotic system can be complex and time-consuming. Therefore, our secondary goal is to train this system following a semi-supervised approach with Generative Adversarial Networks (GANs), so we can reduce the amount of labeled data required. We show in experimentation that our methodology works for deformable objects, which are challenging because they produce dissimilar responses when compared to solid items.

*This work was supported in part by the Spanish Government and the FEDER Funds [BES-2016-078290] and in part by the European Commission [COMMANDIA SOE2/P1/F0638], action supported by Interreg-V Sudoe.

[1]Brayan S. Zapata-Impata and Pablo Gil are with Automatics, Robotics and Artificial Vision Lab, Institute for Computer Research, University of Alicante, Ctra. San Vicente del Raspeig s/n, 03690, Alicante, Spain {brayan.impata, pablo.gil}@ua.es

II. RELATED WORKS

The problem of generating robotic tactile responses from visual perception has not been widely covered in the literature. Recently, Hogan *et al.* [2] presented a re-grasping strategy for a robotic gripper equipped with two GelSlim tactile sensors. In their proposal, the robot first contacted a target object in order to register an initial pair of real tactile images. Then, synthetic tactile images were generated by producing translations on various directions of the images. Although their approach showed great results, it still needed an initial real contact in order to work, since the generated tactile responses were conditioned by this contact.

In [3], Lee *et al.* presented a work in which visual and tactile data were generated, one from the other, using GANs. The authors used a dataset which contained: i) close pictures of pieces of fabric taken with a digital camera and ii) tactile images which registered the response of the sensor upon contact with the pieces of fabric. The authors trained two generator networks: one produced tactile images using an input picture of the touched piece of fabric, whereas the other produced pictures of the fabric under contact in a given tactile image. In this fashion, Li *et al.* [4] trained two GAN-based systems using auto-encoders for a similar task. However, their dataset was composed of sequences of pictures and tactile images which were recorded through touch sequences of a single sensor with various objects.

These two works used touches of a single sensor over objects. We find it problematic to generalize from this setting to grasps. A single touch can produce different tactile responses depending on the orientation of the manipulator, the object and the supporting surface (i.e. table). For example, a touch on a specific spot of a lying cylinder executed perpendicularly to the table – touch and table apply opposite, collinear forces – probably produces a different response to exactly the same touch over the same spot if the object is standing and, therefore, the touch is executed in parallel to the table – the forces are no longer collinear.

Recently, Tian *et al.* [5] proposed a tactile-based model predictive control for carrying out three manipulation tasks with three solid objects using a GelSight-like sensor. In their proposal, a video prediction architecture is used for producing sequences of artificial tactile images, using as input an initial real tactile image and a sequence of candidate actions. After training, their system produced tactile images that matched the manipulation actions. However, it still needed an initial real contact in order to work.

Workshop on Robotic Manipulation of Deformable Objects (ROMADO)
IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2020

978-1-7281-6211-9/20/$31.00 ©2020 IEEE

We propose a novel way to approach this problem in order to overcome the limitations of previous works. In addition, we demonstrate its effectiveness on deformable objects, which are usually not considered in the aforementioned literature. In short, we propose using a Semi-supervised Regression GAN (SR-GAN) [6] for training a discriminator network that regresses tactile responses from 3D point clouds of objects and target grasping data. Besides, we train a generator that produces clouds which are similar to the real ones in the feature space learned by the discriminator. Furthermore, semi-supervised learning is applied in order to reduce the requirements of labeled data. This solution is built on top of a previous work [7], being the contributions of this work as follows:

- We improve our system by training a GAN in which the generator produces fake samples that contribute to the training of the tactile generation system.
- We reduce the number of labeled samples in the training set using a semi-supervised learning methodology. We demonstrate that this improves the performance of the tactile generation system.

III. PROPOSAL

Our goal is to generate the tactile readings that would be registered if a grasp were executed on a deformable object. To this end, we propose using a 3D point cloud of the deformable object to be grasped and data related to the target grasp. In order to approach this regression task, semi-supervised learning and GAN are utilized.

A. *ROBOTIC SYSTEM*

The tactile sensor used in this work is the BioTac SP sensor developed by SynTouch [8]. This tactile sensor is made of an elastic skin and it holds 24 electrodes and 4 emitters distributed on an internal core. There is a fluid located between the skin and the internal core as well. During a contact, forces applied on the external skin produce displacements of the internal fluid and, therefore, the electrodes read different voltage values. Additionally, the sensor has a global pressure sensor located in its base, which experiences forces from the displacement of the fluid.

We equipped two sensors on the thumb and middle finger of a Shadow Dexterous Hand [9]. In order to capture visual data, an Intel RealSense D415 depth camera was installed in an eye-to-hand configuration, so its point of view provided data from the top of our workspace. Hence, we work with partial views of the objects.

B. *INPUT VISUAL REPRESENTATION*

We use 3D point clouds as a stimulus to produce tactile data. This type of data structure represents the geometry of the object in a better way than a 2D image because it encodes curvature and volume information, among others. These attributes should be of great importance for learning to estimate tactile responses: areas with similar geometrical shapes should produce similar responses on the sensors.

In addition, we include target grasp data in the input to our system. These data are composed of two 3D-located grasping points and the rotation the hand should acquire to reach those points. Consequently, the input for our system can be obtained from visual data, allowing us to generate tactile responses without initial contacts nor moving the robot.

C. *OUTPUT TACTILE DATA*

The BioTac SP sensors provide two sources of tactile responses: an estimation of the general pressure, which we denote as PDC; and a more precise estimation registered by its 24 electrodes, which we denote as $E = \{e_1, e_2, ..., e_{24}\}$. Since there is a sensor at the thumb TH and another at the middle finger MF, there are four target responses that we can generate: PDC_{TH}, PDC_{MF}, E_{TH} and E_{MF}.

The values provided by these sources are in custom discrete units: the BioTac SP does not return standard force units, though the relationship of its units with Newtons has been studied before [10]. As a result, we develop our work with these custom units, keeping in mind that a higher value means a lower pressure experienced by the sensor. Besides, every data modality (PDC_{TH}, PDC_{MF}, E_{TH}, E_{MF}) produces readings in distinct ranges of values due to differences in the amount of liquid inside the sensor itself. The discrete ranges of values found empirically for these tactile modalities and our sensors are: $PDC_{TH} \in [2500, 3400]$, $PDC_{MF} \in [1600, 2300]$, $E_{TH} \in [100, 3600]$ and $E_{MF} \in [500, 3800]$. Consequently, we propose learning to generate each tactile source independently, instead of mixing them for training a single model.

D. *SEMI-SUPERVISED REGRESSION GAN*

The problem defined in this work is a regression task. Our goal is to develop a system that learns to regress tactile responses. Regularly, we would record a dataset of inputs and resulting regression values, in order to generate a set of experiences which can be used for training a supervised model. However, gathering pairs of input data and resulting tactile responses is costly: acquiring a single labeled sample requires commanding a robotic system, which takes time to move. Moreover, when dealing with robots, there is larger number of variables that need to be controlled and, therefore, there are more potential points for failure.

In consequence, we propose using a semi-supervised methodology for training a regression system. We are going to train our system with a dataset of labeled samples along with another dataset of unlabeled samples. The labeled dataset holds object's clouds, target grasp data and real tactile responses whereas the unlabeled set is only composed of object's clouds and target grasp data. In addition, a GAN is used so we can learn to generate fake inputs for our system and, therefore, increase further the number of inputs for training.

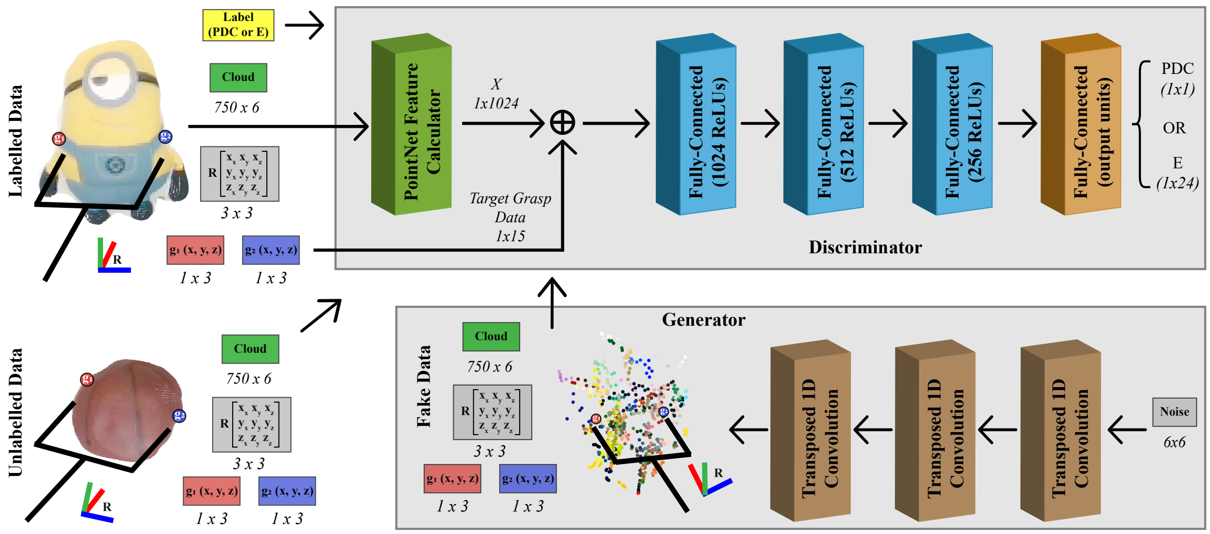

Fig. 1. Architecture implemented for carrying out the semi-supervised regression of tactile responses using point clouds and target grasp data. It is based on a discriminator trained with labeled and unlabeled data as well as fake data from a generator network.

We propose a modification of the SR-GAN [6] for building our system. The idea behind the SR-GAN is to train a generator $\mathcal{G}$ which generates data that match the distribution of the real data, either labeled or unlabeled. However, its goal is not generating realistic samples but rather data that have similar features to those of the real samples. That is: the generator $\mathcal{G}$ does not create samples that match the real data distribution on the original space but on the feature space learned by the discriminator $\mathcal{D}$.

In more detail, we train our discriminator with the following loss functions:

$$\begin{aligned} L_{\mathcal{D}} =& L_{supervised} + L_{unsupervised} \\ L_{supervised} =& w_L * L_{labeled} \\ L_{unsupervised} =& w_U * L_{unlabeled} + w_F * L_{fake} \end{aligned} \tag{1}$$

$$L_{labeled} = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}} \tag{2}$$

$$L_{unlabeled} = \sum_{i=1}^{n} \frac{|f(x_{L_i}) - f(x_{U_i})|^2}{n} \tag{3}$$

$$L_{fake} = \sum_{i=1}^{n} \frac{|f(x_{L_i}) - f(x_{U_i})|^2}{n} \tag{4}$$

where w_L, w_U, w_F are a set of weights given to each component of the discriminator loss, y stands for the target regression value (i.e. PDC or E), $\hat{y}$ is the regressed value returned by the discriminator $\mathcal{D}$, $f(x_L), f(x_U), f(x_F)$ are the features calculated by the discriminator $\mathcal{D}$ at a determined layer with the labeled x_L, unlabeled x_U and fake data x_F.

In our work, we have removed the penalty term that was added to the $L_{\mathcal{D}}$ loss in the original SR-GAN work because it did not have significant effects on the convergence of our system. In addition, computing its value was a time bottleneck. Besides, we use the Root Mean Square Error (RMSE) for the $L_{labeled}$ loss instead of the one proposed in [6]. As for the $L_{unlabeled}$ and the L_{fake} loss, we use the ones proposed in the original SR-GAN work.

The $L_{labeled}$ measures the similarity between the output tactile data $\hat{y}$ of the discriminator $\mathcal{D}$ and the real value y of labeled samples. Hence, it is an ordinary supervised loss. The $L_{unlabeled}$ measures the similarity between the labeled samples x_L and the unlabeled ones x_U in the feature space $f(x)$ learned by the discriminator. Consequently, its goal is to learn a feature space in which their differences are reduced. In contrast, L_{fake} is a feature contrasting loss: it tries to make the features of the unlabeled data and the fake data x_F coming from the generator $\mathcal{G}$ as dissimilar as possible.

Regarding the generator $\mathcal{G}$, it is trained using a loss function so that it attempts to generate fake samples x_F which have similar features $f(x_F)$ to those of the unlabeled samples $f(x_U)$:

$$L_{\mathcal{G}} = \sum_{i=1}^{n} \frac{|f(x_{F_i}) - f(x_{U_i})|^2}{n} \tag{5}$$

E. NETWORK ARCHITECTURE

The architecture of our discriminator network $\mathcal{D}$ and the generator network $\mathcal{G}$ are shown in Fig. 1. The discriminator $\mathcal{D}$ is based on PointNet [11], which is a neural network that computes features over point clouds. This network generates a descriptor X with 1024 features using convolutions, which is convenient for our work: we use as input an object's cloud $\mathbb{C}$ and the network calculates a feature vector $X = f(\mathbb{C})$. Consequently, we set the output of this intermediate layer, denoted as PointNet Feature Calculator, as the target feature space $f(x)$ used in losses $L_{unlabeled}, L_{fake}$ and $L_{\mathcal{G}}$.

Right after the PointNet Feature Calculator layer, we concatenate to the feature vector X the pair of grasping points g_1 and g_2 (3 values per each) and the rotation matrix R representing the pose of the hand (9 values). As a result, the fully-connected layer after the PointNet Feature Calculator receives as input a new vector with 1039 values, which are then used for completing the regression task. We have 3 fully-connected layers with 1024, 512 and 256 Rectified Linear Units (ReLUs), followed by a final layer that outputs 1 value in the case of training with PDC (general pressure value) or 24 values if trained with E (electrodes) data.

In order to produce a fake point cloud $\mathbb{C}_F$, the architecture of the generator $\mathcal{G}$ is based on transposed convolutions. We selected an architecture that outputs clouds of 750×6 points after experimenting with different sizes. This network receives as input a vector $N^{6\times6}$ that has random values sampled from an uniform distribution in the range $[0, 1)$. The vector is processed by 3 layers of transposed 1D convolutions with kernel size equal to 5 and stride equal to 5 in order to generate a fake cloud $\mathbb{C}_F^{750\times6}$ points. Last, the hyperbolic tangent activation function is used.

Finally, the generator network $\mathcal{G}$ only generates the fake point cloud. In order to generate the target grasp data required to train the discriminator network $\mathcal{D}$, we applied the same strategy used for calculating grasps on real clouds, which is described in the next section.

F. DATASET

Using deformable objects, we recorded a dataset of labeled samples which contain the object's cloud and the target grasp data. Additionally, we executed with the robot the computed grasps in order to record as well the resulting tactile responses. Thereby, a sample is a tuple $\Theta = \langle \mathbb{C}, g_1, g_2, R, PDC_{TH}, PDC_{MF}, E_{TH}, E_{MF} \rangle$.

Grasps were calculated using GeoGrasp [12]. This method calculates pairs of contact points using 3D point clouds with partial views of objects. Grasps returned by GeoGrasp tend to be near the centroid of the object's cloud and perpendicular to its principal axis. In order to produce a greater variability of grasps throughout the surface of the objects, we modified GeoGrasp so it would randomly move the contacts along an approximation of the object's axis. As a result, grasps were distributed and the variability of the dataset was augmented.

We executed grasps on 4 objects (see Fig. 2 (left)): a stuffed Minion, a flat ball, a sponge and a stuffed volleyball. They were chosen because they have different shapes, textures, stiffness and materials (e.g. foam, synthetic fibre). Since they are all soft, they produced varying tactile responses, as can be seen in Fig. 2 (right). 1000 grasp samples were recorded for each object, so the dataset contains 4000 samples in total.

IV. EXPERIMENTATION

We create a test set extracting 200 samples per object from the whole dataset of 4000 samples. Therefore, the test set contains 800 samples, keeping the objects equally represented. 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.

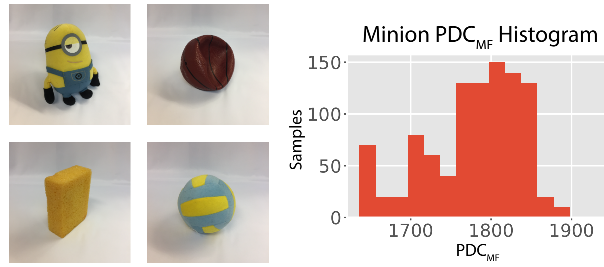

Fig. 2. Dataset collected: objects (left) and example of PDC_{TH} responses registered for one of them (right).

It is possible to transform a labeled sample into a unlabeled sample by omitting the tactile responses $(PDC_{TH}, PDC_{MF}, E_{TH}, E_{MF})$ in the sample tuple Θ. We apply this to create 5 training sets from the 3200 labeled samples with different ratios of labeled/unlabeled samples:

- *T10* is a training set in which 10% of the 3200 samples are unlabeled. Therefore, it contains 2880 labeled samples and 320 unlabeled samples.
- *T30* is a training set in which 30% of the 3200 samples are unlabeled. Therefore, it contains 2240 labeled samples and 960 unlabeled samples.
- *T50* is a training set in which 50% of the 3200 samples are unlabeled. Therefore, it contains 1600 labeled samples and 1600 unlabeled samples.
- *T70* is a training set in which 70% of the 3200 samples are unlabeled. Therefore, it contains 960 labeled samples and 2240 unlabeled samples.
- *T90* is a training set in which 90% of the 3200 samples are unlabeled. Therefore, it contains 320 labeled samples and 2880 unlabeled samples.

We experiment with these ratios of labeled/unlabeled samples for training our SR-GAN. We compare its performance with a baseline method in order to check if adding unlabeled samples and fake samples improve the performance of the system. This baseline is a network whose architecture is exactly the same to that of the discriminator in the SR-GAN. That is: it is a regular PointNet with extra fully-connected layers for performing the regression task. However, it is only trained with the labeled samples from each training set and its loss function is just composed by $L_{labeled}$ (see (2)).

We execute 5 independent runs of training on one of the training sets and then evaluating on the test set of 800 samples. This is repeated for each training set with out SR-GAN and the baseline PointNet. This experimentation methodology was chosen to provide average performance measurements with error values. For measuring the networks performance on this regression task we use the RMSE.

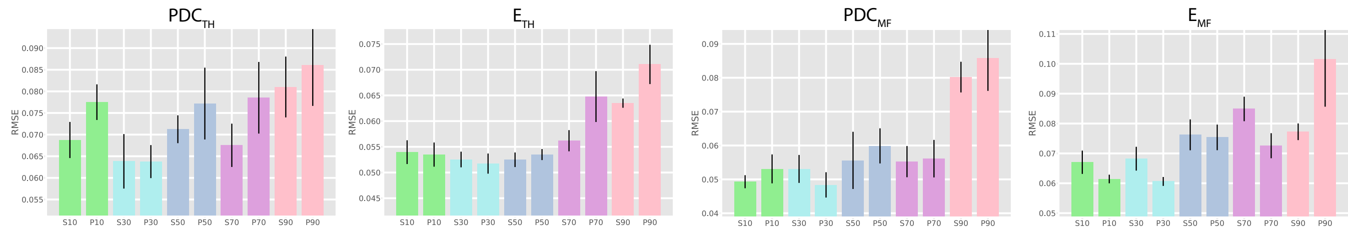

Fig. 3. Average RMSE from 5 runs of training on 3200 samples and testing on 800 samples for each tactile modality. *SX* stands for a SR-GAN trained with a *X* percentage of unlabeled samples (e.g. *S10* is a SR-GAN trained with 2880 labeled and 320 unlabeled samples). *PX* stands for a baseline PointNet trained with *X* percentage less samples from the 3200 training samples (e.g. *P10* is a PointNet trained with 2880 labeled samples).

Finally, we fixed the discriminator $\mathcal{D}$ loss weights to $w_L = 1, w_U = 10, w_F = 0.1$ after initial experimentation with their values. The SR-GAN was trained with learning rate equal to 0.01, batch size equal to 10 and 30 epochs. The baseline PointNet was trained with identical learning rate, batch size equal to 32 and 50 epochs. The Adam optimizer was used.

We show in Fig. 3 the results obtained from training the SR-GAN proposal and the baseline PointNet on the 5 training sets and later evaluating on the test set. Note that results for the SR-GAN are shown in bars labeled as S, whereas the baseline PointNet is denoted as P. The numbers to the right of the labels denote the training set used: for example, $P30$ is a baseline PointNet trained on the 2240 labeled samples of training set *T30*.

For each tactile modality, we have 10 results: 5 belong to the SR-GAN and the other 5 belong to the baseline PointNet. We can check the effect of learning with unlabeled and fake samples by comparing the results of both models on the same training set. We can do this because both models are trained on exactly the same labeled samples. The only difference is that the SR-GAN also receives the unlabeled samples and its generator produces more inputs creating fake clouds.

In general terms, it can be seen that for both models the error increases as we train on less labeled samples. For example, the RMSE on the generation of PDC_{TH} values is greater than 0.08 points for both models when trained on the *T90* set (check bars *S90* and *P90* on the PDC_{TH} plot in the figure), but it is lower when the models are trained on the other training sets.

The SR-GAN yields lower average error values than the baseline PointNet in 4/5 experiments on PDC_{TH} data. When the SR-GAN yields a lower test error, it is at least 6.2% lower (see results on *T90*) and this decrease peaks at 16.2% (see results on *T70*). In contrast, the experiment in which the PointNet obtains a lower error, *S30* yields an error only 0.09% higher than *P30*.

The proposed system also improves the baseline performance in 3/5 experiments on E_{TH} data. Its error is at least 1.9% lower (see results on *T50*) and it peaks at 15.2% improvement (see results on *T70*). However, the worst error obtained by the SR-GAN on this data when compared to the baseline PointNet is only 1.5% higher (see results on *T30*).

On PDC_{MF} data, the SR-GAN obtains lower error than the baseline in 4/5 experiments. Its error is at least 1.5% lower (see results on *T70*) and this improvement peaks at 7.7% (see results on *T50*). However, the worst error obtained by the SR-GAN when compared to the baseline PointNet is 8.8% higher (see results on *T30*).

For the previously analyzed modalities, the proposed SR-GAN consistently improves on the baseline PointNet on at least 3 out of the 5 experiments carried out. However, for the E_{MF} data, it only improved the baseline results on 1/5 experiments. Although the SR-GAN obtained a 31.5% lower error rate in that case (see results on *T90*), it yielded error rates which were between 1.1% and 14.5% higher on the rest of the tests (see results on *T50* and *T70*).

Last, we want to observe that errors are low if we compare them with the ranges of values of the tactile modalities. Lowest PDC_{TH} error was obtained by PointNet trained with *T30*. This error equals 6.37% RMSE, which are 57 points if we convert them back to the range of this data modality ($[2500, 3400]$). As for E_{TH}, the lowest RMSE was obtained by PointNet trained with *T30*. It equals 5.17%, which are 181 points in the range of this modality ($[100, 3600]$). In the case of PDC_{MF}, the lowest RMSE was obtained by PointNet trained with *T30* and it equals 4.83%. This value equals 34 points in the range of this modality ($[1600, 2300]$). Regarding E_{MF}, the lowest RMSE was obtained by PointNet trained with *T30* and equals 6.06%, which are 200 points in the range of this modality ($[500, 3800]$).

V. CONCLUSIONS

We introduce in this work our early results on prediction of tactile perception from vision on deformable objects. We propose for this regression task the use of a SR-GAN: a model that carries out a semi-supervised training of a GAN for regressing tactile data. We train our system with samples of point clouds, target grasps and the resulting tactile responses. In addition, we also use two types of unlabeled samples: real point clouds with target grasps and fake clouds with target grasps.

In experimentation with deformable objects, we find that using unlabeled samples can improve the performance of the system, when compared to a baseline method only trained with labeled samples. This performance is more likely to happen if fewer labeled samples are available.

Workshop on Robotic Manipulation of Deformable Objects (ROMADO)
IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2020

Our conclusions are limited by the fact that we are only running these experiments with 4 objects and 4000 samples. It can be seen that SR-GAN is not consistently improving the performance of the baseline method. This is probably a consequence of lacking further experimentation with larger sets. Next steps will be taken for adding more samples to the datasets.

In addition, we want to perform further experiments with deformable objects which are similar in appearance but have different stiffness. Deformable objects are hard for this task because their looks might not directly relate to their stiffness degree, which is a property that directly affects our tactile prediction.

REFERENCES

[1] R. W. Fleming, "Visual perception of materials and their properties," *Vision Research*, vol. 94, pp. 62–75, 2014.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] G. Olmschenk, Z. Zhu, and H. Tang, "Generalizing semi-supervised generative adversarial networks to regression using feature contrasting," *Computer Vision and Image Understanding*, pp. 1–19., jul 2019.

[7] B. S. Zapata-Impata, P. Gil, Y. Mezouar, and F. Torres, "Generation of Tactile Data from 3D Vision and Target Robotic Grasps," *IEEE Transactions on Haptics*, pp. 1–11, 2020.

[8] Syntouch, "BioTac SP," last accessed: 09-08-2020. [Online]. Available: https://syntouchinc.com/technology/

[9] Shadow Company, "Shadow Dexterous Hand," last accessed: 09-08-2020. [Online]. Available: http://www.shadowrobot.com/products/dexterous-hand/

[10] 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.

[11] 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.

[12] 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.