Touching a NeRF: Leveraging Neural Radiance Fields for Tactile Sensory Data Generation

Anonymous Author(s)
Affiliation
Address
email

Abstract: Tactile perception has a key role to play in robotics applications such as manipulation and exploration. However, manually touching objects to collect tactile data is time-consuming, especially when compared to visual data acquisition. This constraint limits the use of tactile data in machine learning solutions to robotics applications. In this paper, we propose a generative model to simulate realistic tactile sensory data for use in downstream robotics tasks. Starting with easily-obtained camera data, we train Neural Radiance Fields (NeRF) for objects of interest. We then use the rendered RGB-D images from the NeRFs as inputs to a conditional Generative Adversarial Network model (cGAN). By leveraging a NeRF model, we can render images of a scene from arbitrary view angles. This in turn allows us to generate tactile images from arbitrary approach angles by conditioning on the outputs of NeRF models. We evaluate the generated data quantitatively using metrics such as the Structural Similarity Index and Mean Squared Error. The generated tactile data are further evaluated in a downstream tactile classification task both in simulation and on a real robot arm. The experimental results show that the generated dataset is able to increase classification accuracy by around 10%, demonstrating the potential for the proposed approach to generate realistic tactile sensory data and augment manually collected tactile datasets. In addition, we demonstrate that our model is able to transfer from one tactile sensor to another with a small fine-tuning dataset.

Keywords: Camera-based tactile sensing, cross-modal tactile data generation

1 Introduction

Humans rely heavily on tactile sensing for tasks such as identifying and grasping objects (e.g. retrieving keys from a pocket) [1, 2]. In this context, tactile sensing is fundamental to retrieve contact information or properties of the object such as roughness or stiffness, and is also able to complement vision in occluded scenarios [1]. Tactile sensing has proven to be fundamental in robotics applications such as manipulation and control [3] and object or texture recognition [4]. These tasks, especially the ones related to tactile-based object recognition, are usually tackled with machine learning methods that typically require large amounts of data for training [1, 5, 6]. However, collecting tactile data is challenging as it requires the robot to physically interact with the environment of the object of interest. While cameras can capture the global shape of an object, tactile sensors can only capture local features. Therefore, a long and time-consuming exploration procedure is usually required to capture the whole shape [7]. Furthermore, beyond the problem related to the tactile exploration, it must also be considered that tactile sensing is still lacking of standards at hardware level [8]. This means that for the same physical stimulus, the output of different tactile systems can differ, thus limiting the validity of the collected data to a specific sensing technology.

In view of the difficulties with tactile data collection, the problem of generating synthetic tactile sensor responses from data acquired using different modalities (which are easier to collect) becomes

Submitted to the 6th Conference on Robot Learning (CoRL 2022). Do not distribute.

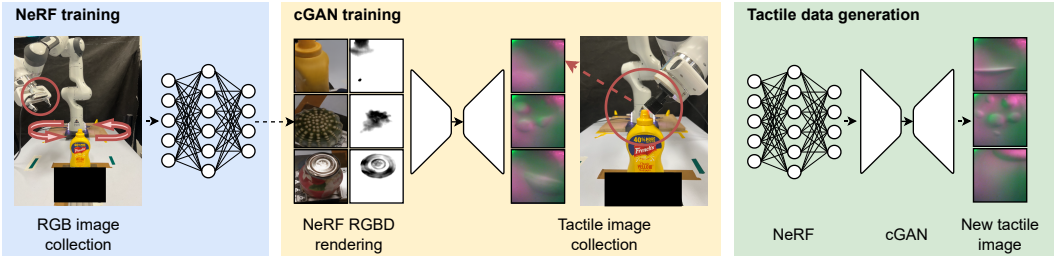

Figure 1: Overview of the framework. We first capture camera images around the object to train a NeRF model. We then collect a tactile image dataset to train a conditional GAN model that takes NeRF rendered RGB-D images as input and outputs tactile images. Once the cGAN model is trained, we can query NeRF models with arbitrary poses, and pass the RGB-D output to the cGAN model to obtain the desired tactile image from the same orientation as the query pose.

relevant. In particular, recent works show that vision data (RGB-D images) contain rich sensory information that can be used to generate tactile data [9, 10, 11]. Given a camera or depth image of the object surface as an input, these approaches can generate the corresponding tactile sensor output. One limitation of these vision-based generative approaches is that the models require the collection of visual samples at given positions and orientations to generate the corresponding synthetic tactile data [9]. However, with the development of neural volume-rendering techniques such as NeRF [12], it is now possible to synthesise high quality RGB-D images for novel view orientations of a scene, given only sample 2D images and their associated camera poses [12]. In this way, NeRF provides additional information of the structure of scene that we leverage for generation of tactile images for 3D objects.

The contribution of this paper is to introduce a novel framework for leveraging the RGB-D images that NeRFs render to generate tactile sensory data via a deep generative model. As far as the authors are aware, ours is the first work that learns a generative model conditioned on RGB-D inputs for the generation of tactile data. Compared to simulation-based approaches for tactile data generation, our framework removes the need for accurate hand-engineered modelling and calibration of the tactile sensor as well as the objects of interest. Compared to other learning-based approaches, our method is capable of handling 3D objects of different geometries, and is able to generalise to novel views of the object through the use of NeRF models. The use of NeRF models also removes the need for a RGB-D camera to manually obtain a new input image each time a new object reading is desired. Further, by using a deep generative model conditioned on a sensor background image, our framework is capable of transferring to a new sensor with a small fine-tuning dataset, overcoming the domain gap between different tactile sensors and opening up the potential to leverage different tactile datasets for improved performance.

2 Literature Review

This section reviews the approaches that simulate or generate data for camera-based tactile sensors [13, 14]. These devices use a camera to capture the deformation of a soft medium and output a *tactile image*. The majority of the works presented in the literature address the problem of generating tactile data from vision, using camera-based sensors, as both modalities encode the information using the same data structure [9, 10, 11, 15]. Other approaches are based on the development of accurate simulators for camera-based tactile sensors, which aim to make the robot learn tactile features in a simulated environment and transfer the knowledge to the real world [16, 17, 18, 19]. For example, the TACTO simulator presented in [16] allows one to emulate both Digit and the OmniTact sensors by simulating contacts in an off-the-shelf physics engine [14, 20]. Different simulators, such as Taxim aim to simulate the responses of a GelSight sensor by using a polynomial look-up table [17]. Another approach is to leverage methods such as physics-based rendering [18] and depth-maps from physics simulators [19]. Additional works attempt to address the sim2real issue in these tactile simulators through the use of CyclyGANs [21] and texture generation networks [22]. Even though tactile

simulators present an attractive option for data synthesis, these tactile simulators require accurate modelling and calibration of the specific tactile sensor via additional calibration equipment [16, 17]. An accurate object model is also needed for simulating contacts [17]. In contrasts, our proposed learning-based framework removes the need for modelling the sensor or the object.

A separate class of work explores the use of learning-based approaches to generate tactile data. For example, Li et al. [10] estimate the response of a GelSight sensor using a cGAN model [23] conditioned on a vision sequence, which captures the robot touching the object. However, their method requires a robot to perform the touching action to generate the tactile response. Patel et al. [15] leverage depth sensors and the object mesh to generate tactile images, similarly building on a cGAN model. Gao et al. [24, 25] also attempt to render tactile images of individual objects directly using a NeRF-like model. However, their approach is incapable of generalising to new objects because a new NeRF model needs to be trained on tactile images for the new object. Lee et al. [9] employ a cGAN to generate tactile images from vision, and vice-versa, on a dataset containing top-down views of flat clothes – our work is most similar in spirit to this. However, we differ significantly in our use of RGB-D images rendered from NeRF models. This removes the need to retake a camera image for each desired tactile image. Our problem setting is also harder, as it focuses on 3D objects with complex geometries. In addition, we demonstrate the transfer of the trained cGAN model to generate data for a new tactile sensor, which, as far as the authors are aware, is the first that demonstrates such generalisation capability across tactile sensors.

3 Methods

As seen from Figure 1, our approach involves first training individual NeRF models for objects of interest. Although they only require RGB images for training, the use of NeRFs enables us to learn the 3D structure of the scene and render RGB-D images from arbitrary view points. With these RGB-D images, we propose to train a cGAN model to generate tactile data conditioned on both the RGB-D images rendered by NeRF models and a reference background image for the tactile sensor. To generate tactile data from novel poses, we simply render corresponding RGB-D images from an object's NeRF model and pass the results as inputs to the cGAN model.

3.1 Neural radiance fields

A Neural Radiance Field (NeRF) is a model for synthesising novel views of scenes [12]. It requires a set of RGB images and their associated camera poses for training. A trained NeRF is able to synthesise high-quality and novel views of the modelled scene. Formally, a NeRF model represents a continuous scene as a function F, which takes the 3D location of a point $\mathbf{m} = (x, y, z)$ and a 2D viewing direction $\mathbf{d} = (\theta, \phi)$ as input, and predicts the RGB colour $\mathbf{c} = (r, g, b)$ and volume density σ of the point from the particular viewing direction. The learned function is parameterised using a multi-layered perceptron F_Θ. Using F_Θ, images of the scene can be rendered from arbitrary camera poses. Given the camera center $\mathbf{o}$, the color of each pixel is estimated by summing N samples along the camera ray $C(\mathbf{r})$, where $C(\mathbf{r}) = \mathbf{o} + t\mathbf{d}$, using a numerical quadrature approximation [26]:

$$\hat{C}(\mathbf{r}) = \sum_{n=1}^{N} T_n(1 - \exp(-\sigma_n \delta_n))\mathbf{c}_n \tag{1}$$

where $T_n = \exp(-\sum_{n'=1}^{n-1} \sigma_{n'}\delta_{n'})$ and $\delta_n = t_{n+1} - t_n$. By predicting the volume density of sampled points in this process, a NeRF can trivially render the depth image of the scene [12], a capacity that we take advantage of in this study. After rendering the required RGB-D image, we perform simple preprocessing by centre cropping of the RGB-D image and normalisation of the depth image before passing the RGB-D image as input to the cGAN model.

3.2 Conditional generative adversarial network

After using the NeRF model to render the RGB-D image of an object from a target camera pose $(\mathbf{o}, \mathbf{d}_c)$, the goal is to generate the corresponding tactile image. We define the matching criteria

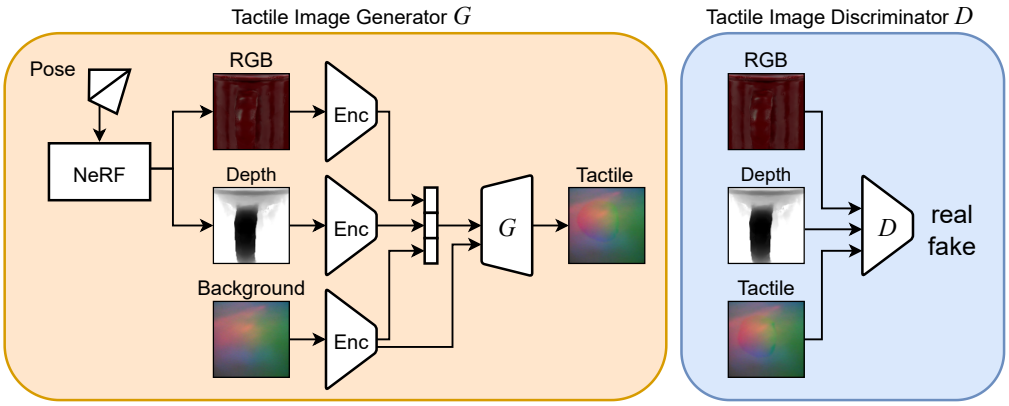

Figure 2: Proposed cGAN architecture. Given a desired pose, an RGB-D image is rendered from the NeRF model. The RGB, depth, and reference sensor background images are then encoded through separate encoders. The output of these encoders are concatenated before being fed to the generator network to generate the desired tactile image. During training, a classifier is used to differentiate whether a tacile image is generated by the generator (fake) or is a ground-truth tactile image (real). The classifier loss and an L1 image loss are used to train the generator during training.

where the tactile sensor orientation is the same as the camera orientation, and the tactile sensor position is at a point along the centre camera ray $(\mathbf{s}, \mathbf{d}_c)$, where $\mathbf{s} = \mathbf{o} + t_s \mathbf{d}_c$ and t_s is dependent on the geometry of the object.

We employ a conditional Generative Adversarial Network to learn the mapping $G : \{\mathbf{x}, \mathbf{z}\} \rightarrow \mathbf{y}$ between a rendered RGB-D image $\mathbf{x}$ and noise $\mathbf{z}$, on one hand, and the corresponding tactile image $\mathbf{y}$, on the other [27]. Taking inspiration from prior work [10], we modify the cGAN model to condition the generative model on both the RGB-D image and a reference background image $\mathbf{b}$, with skip-connections added between the encoder for the reference background image and the generator (see Figure 2 for detailed architecture). Following the cGAN literature [27], the noise $\mathbf{z}$ can be implemented using Dropout [28] in the generator network G. The objective function for the cGAN model is

$$G^* = \arg\min_G \max_D \mathcal{L}_{\text{cGAN}}(G, D) + \lambda \mathcal{L}_{\text{L1}}(G) \tag{2}$$

where

$$\mathcal{L}_{\text{cGAN}}(G, D) = \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p(\mathbf{x},\mathbf{y})}[\log D(\mathbf{x}, \mathbf{y})] + \mathbb{E}_{\mathbf{x},\mathbf{z}\sim p(\mathbf{x},\mathbf{z})}[\log(1 - D(\mathbf{x}, G(\mathbf{x}, \mathbf{z})))] \tag{3}$$

and

$$\mathcal{L}_{\text{L1}}(G) = \mathbb{E}_{\mathbf{x},\mathbf{y},\mathbf{z}\sim p(\mathbf{x},\mathbf{y},\mathbf{z})}[\|\mathbf{y} - G(\mathbf{x}, \mathbf{z})\|_1] \tag{4}$$

4 Experiments

Experiments are first performed in simulation as proof-of-concept and then in the real-world to test the hypothesis that generated tactile data can improve performance on down-stream robotics tasks. To collect tactile images, we employ Digit tactile sensors [14]. For the simulation experiments, we use TACTO to simulate the Digit tactile readings [16]. We note that the TACTO simulator uses PyBullet as its physics engine and PyRender as the rendering engine.

4.1 Dataset generation

Simulation For the simulation experiments, we select 27 common household objects from the scanned YCB dataset [29]. To train a NeRF model for each object, we collect 48 images taken from evenly sampled poses around a sphere centered on the object. To perform a realistic collection of tactile readings, we randomly sample initial positions on a sphere centred on the object and move the tactile sensor from the initial position towards the centre of the object. To obtain tactile

readings, a constant force is applied to the simulated Digit sensor, while the objects are fixed. We discard cases when no reading is obtained, and collect 500 touches per object. Correspondingly, 500 RGB-D images are synthesised from the NeRF model for each object, using the sensor orientation as the view orientation. The rendered RGB-D images are then preprocessed and paired with their corresponding tactile images for training the cGAN model. For testing and evaluation, we hold out 3 distinct objects as the novel object test set and use 24 objects for training. We also collect a validation set and a novel view test set of 50 paired tactile and RGB-D images from novel orientations for each training object.

Real-world For experiments in the real-world, we select 8 common household objects from both the YCB dataset and other objects belonging to similar categories [29]. For training the NeRF models, we collect 118 images taken from camera poses that are uniformly sampled within a range in the hemisphere around the object using the set-up in Figure 1.

To automate data collection, we use a 7-DoF Franka Emika Panda arm with an Intel RealSense D415 camera for taking images, and select the camera pose range based on the workspace of the robot. For collecting the tactile reading, as seen in Figure 1, we attach a Digit sensor on the end effector flange of the Panda arm, and control the arm to move towards the centre of the object. For training, we perform 132 touches per object and collect 50 frames per touch. We also discard the cases when no reading is obtained. We then synthesise and preprocess the corresponding RGB-D images using the NeRF model and pair with the tactile images for training the cGAN model. For testing, we hold out 2 distinct objects as the novel object test set, and we hold out 12.5% of the training dataset as the novel view test set. It should be noted that the test set is randomly chosen using the index of the touch and not the frame. The resultant training dataset contains 398 touches and 19,900 frames.

4.2 Implementation details

For training the NeRF models for objects in simulation and in the real-world, we employ the default configurations in the original NeRF paper [12]. For the cGAN model, we use a U-Net architecture that employs Conv-Norm-ReLU blocks as both encoder and decoder layers [30]. Both encoder and decoder use 7 layers of the convolutional blocks. RGB, depth, and reference background images are encoded using separate decoders that have the same architecture, where depth is treated as grayscale. Please see the Appendix for further details.

4.3 Evaluation details

As losses in cGAN are not directly indicative of the quality of the training results, it is difficult to evaluate the realism of the generated tactile images. It is also difficult for a human to evaluate the realism of such tactile images. Thus, we employ a set of approximate metrics and an example downstream task to evaluate the quality of the generated images. We first evaluate the Structural Similarity Index (SSIM) and the Mean Squared Error (MSE) between the generated image and the ground truth tactile image on the hold out test sets. Note that the average SSIM across RGB channels is taken, and that in simulation, ground-truth images refer to the simulated tactile images from the TACTO simulator. Next, we use tactile classification as an example task to evaluate the usefulness of the generated tactile dataset for robotic tasks. The same evaluation pipelines are applied to results in both simulation and real-world experiments. In evaluation, we scale all images to 128×128 which is also the size of the output of the cGAN model.

For both simulation and real-world experiments, we evaluate the generated images in a tactile classification task, using a simple convolutional neural network to perform object classification based on one tactile image. For comparison, we prepare two sets of training datasets, one consisting of only ground-truth tactile images, the other consisting of both ground-truth and generated image. The test dataset consists of only ground-truth images. The experiment is performed for simulation data and real-world data.

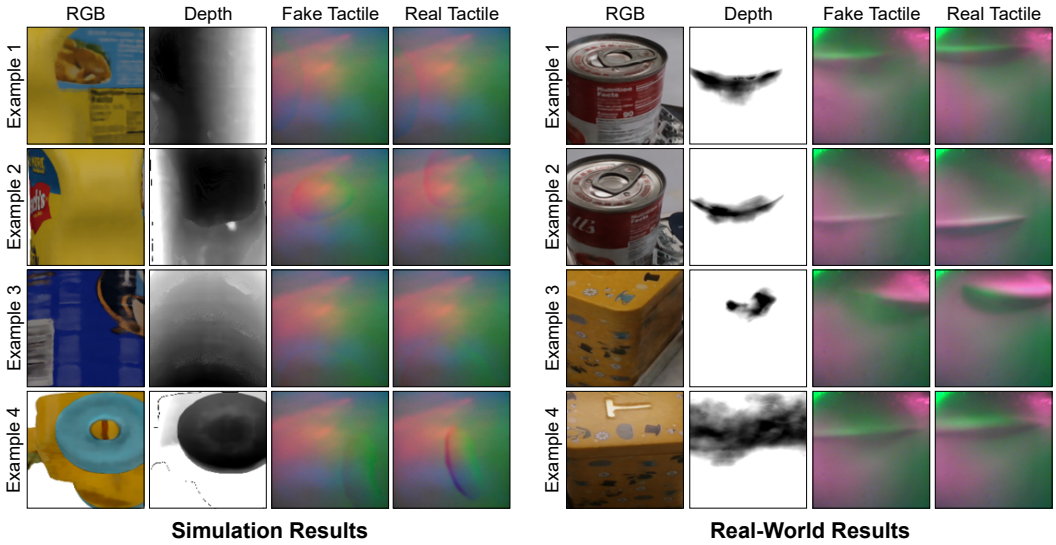

Figure 3: Qualitative results on novel object test sets. Our approach generates faithful tactile images in both simulated (left) and real-world (right) domains on objects it has not seen before in training. The columns show RGB images, depth images, fake tactile images, and real tactile images. The fake tactile images are model generated; the real tactile images provide ground truth.

5 Results and Discussion

We evaluate the results of tactile image generation in both a simulated setting and real-world setting though a set of quantitative metrics and a tactile classification task. We also evaluate the results in transferring the learned cGAN model to a different sensor.

5.1 Simulation results

From Table 1, we can see that our approach is able achieve high SSIM and low MSE values, indicating that the generated tactile images are structurally similar to the ground-truth (simulated) tactile images for both novel views of training objects and novel objects. This is corroborated by the qualitative results as shown in Figure 3. On the novel view dataset, our approach is capable of generating tactile images very similar to ground-truth ones, without having to explicitly model the object or the tactile sensor. The higher MSE error for novel objects could be due to a difference in the position of the predicted indentation, which leads to more differences with the ground-truth pixel-wise.

Test set	SSIM	MSE
Novel view	0.980 ± 0.001	14.5 ± 0.8
Novel object	0.961 ± 0.000	84.5 ± 3
Sensor background	0.921 ± 0.000	67.4 ± 3

(a) SSIM values between generated tactile images and ground-truth (simulated) tactile images. The values shown are average values across the test set with standard errors (rounded to the nearest thousandth).

Dataset	Accuracy
Sim	49 ± 2
Sim + Gen	59 ± 1

(b) Classification results for simulated dataset. Results shown are average values with standard errors.

Table 1: Results of evaluation of the generated tactile image in simulation

To evaluate the quality of the generated tactile images we use tactile object classification as an example task. By augmenting a given tactile dataset with generated tactile images, we are able to improve the success rate of the tactile classifier by a large margin. This shows that the proposed generative model is able to generate tactile images useful for the tactile classification task. Through

leveraging NeRF models, we are also able to render additional RGB-D images of the object from novel view angles and generate tactile images to further augment the dataset with a low cost. The relatively low absolute accuracy of the classification is likely due to the presence of objects with identical shapes (e.g. cups of different colour) in the YCB dataset.

5.1.1 Transferring to a different sensor

By using a much smaller fine-tuning dataset than the original RGB-D and tactile training dataset, we are able to fine-tune a trained cGAN model to adapt to a different sensor with different characteristics. To demonstrate this, we select the OmniTact sensor as an example target tactile sensor for fine-tuning. We collect simulated OmniTact images using a similar procedure as described in Section 4.1 and pair the tactile image with corresponding RGB-D image rendered from NeRF. We collect 5 tactile images for each training object for training and 50 each as the novel view test set. We hold out the same 3 objects as novel object test set. We then continue to train the cGAN model that has been pre-trained on the simulated Digit dataset with the new OmniTact dataset, and evaluate the results using similar metrics as describe in Section 4.3.

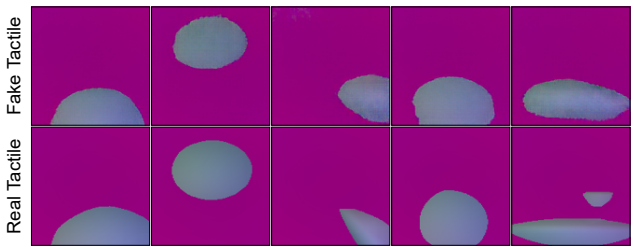

Figure 4: Example of generated tactile images for the simulated OmniTact sensor. In this case, the real tactile images are generated by the simulator.

Model	Test set	SSIM	MSE
Pre-trained (Ours)	Novel view	0.865 ± 0.002	550 ± 10
	Novel object	0.839 ± 0.003	760 ± 30
From scratch	Novel view	0.829 ± 0.002	950 ± 20
	Novel object	0.811 ± 0.005	1300 ± 60
-	Sensor background	0.856 ± 0.001	1390 ± 20

(a) SSIM values between generated tactile images and ground-truth (simulated) tactile images. Results shown are average values with standard errors.

Dataset	Pre-trained	From scratch
Sim	36 ± 2	36 ± 2
Sim+Gen	43 ± 3	26 ± 1

(b) Classification using fine-tuning dataset. Results shown are average values with standard errors.

Table 2: Results of evaluation of the generated tactile image for a new sensor in simulation

Using a much smaller fine-tuning dataset, the generated OmniTact tactile image still resembles the ground-truth (simulated) images, as shown in the SSIM and MSE results in Table 2a and Figure 4. The lower SSIM value for novel objects is likely due to the presence of artefacts in the generated images, which we found could be easily remedied with a small amount of additional fine-tuning data. From Figure 4, it can also be seen that the generated tactile images are able to capture the shape of the indentation. Additionally, the background of the tactile image is accurately replicated, pointing to the benefit of conditioning the generative model on a reference sensor background image. Classification results in Table 2b also demonstrate that the generated tactile images for the new sensor are still useful for increasing the accuracy of the tactile classifier by augmenting the tactile dataset. The fine-tuned cGAN model also outperforms the benchmark cGAN model trained from scratch, demonstrating the usefulness of pre-training and pointing to the potential of the proposed approach in leveraging available tactile datasets for different sensors to improve performance.

5.2 Real-world results

From Figure 3 it can be seen that the proposed framework is able to generate realistic tactile images on both novel views of training objects as well as objects that it has not seen before during training. It

is also able to generate the change in lighting caused by the object pressing against the gelpad. The difference in the amount of indentation between the generated image and the ground-truth image could be due to the variation of the force applied, as we only enforce the force to be within a range.

Table 3 shows that the generated tactile images are able to capture the key properties of ground-truth tactile images taken in the real world. The SSIM values are lower than those in simulation in Table 1, which is to be expected due to the presence of noise in the real-world. Compared to SSIM results reported in previous work ($\sim$0.9) [9], the SSIM values are slightly lower. However, as explained in Section 2, this paper presents a much more challenging setting involving 3D objects with different geometries.

Additionally, classification results in Table 3b indicates that the generate tactile images are useful in increasing the accuracy of the downstream tactile classification task. The higher absolute accuracy compared to the simulated results is likely due to the smaller number of objects for the real-world dataset. Compared to previous work [9], the proposed framework has achieved more significant increase in the accuracy of tactile classification using the generated tactile dataset. This demonstrates the potential of the framework in generating tactile images useful for downstream tasks.

Test set	SSIM	MSE
Novel view	0.845 ± 0.001	64 ± 1
Novel object	0.814 ± 0.001	180 ± 2
Sensor background	0.814 ± 0.000	239 ± 3

(a) SSIM values between generated tactile images and ground-truth (real) tactile images. Results shown are average values with standard errors.

Dataset	Accuracy
Real	73 ± 3
Real + Gen	83 ± 5

(b) Classification results for real-world dataset. Results shown are average values with standard errors.

Table 3: Results of evaluation of the generated tactile image in real-world

6 Limitations

By leveraging NeRF and cGAN models, the proposed framework is able to generate realistic tactile images for an example downstream classification task and remedy the challenges with current approaches. However, the generation of tactile images also poses a couple of challenges. Firstly, we assume a fixed force range and orientation (pitch and yaw) of the tactile sensor. The generalisation to different force and orientations is left to future work. Secondly, a NeRF model needs to be trained for each new object, which can be time-consuming. However, recent advances in the acceleration of NeRF training mitigate this significantly [31].

7 Conclusion

In conclusion, we present a novel framework for the generation of tactile images for camera-based tactile sensors. The framework leverages NeRF models to render RGB-D images of an object from a desired pose, and pass the RGB-D image to a cGAN model to generate the desired tactile image. Results in both simulation and real-world demonstrate that the generated tactile images are structurally similar to ground-truth images. The cGAN model is also able to pre-train on tactile datasets from another sensor and transfer to a new sensor with a small fine-tuning dataset. Through using tactile classification as a demonstration task, we show that the generated images can be used to augment tactile datasets for downstream robotics tasks. Future work should be focused on further conditioning the cGAN model on force and the orientation of the tactile sensor with respect to the object.

References

[1] More than a feeling: Learning to grasp and regrasp using vision and touch. *IEEE Robotics and Automation Letters*, 3:3300–3307, 2018.

[2] Y. Narang, B. Sundaralingam, M. Macklin, A. Mousavian, and D. Fox. Sim-to-real for robotic tactile sensing via physics-based simulation and learned latent projections. *arXiv preprint arXiv:2103.16747v1*, 2021.

[3] Z. Kappassov, J.-A. Corrales, and V. Perdereau. Tactile sensing in dexterous robot hands – review. *Robotics and Autonomous Systems*, 74:195–220, 2015.

[4] S. Luo, J. Bimbo, R. Dahiya, and H. Liu. Robotic tactile perception of object properties: A review. *Mechatronics*, 48:54–67, 2017.

[5] J. M. Gandarias, J. M. Gómez-de Gabriel, and A. J. García-Cerezo. Enhancing perception with tactile object recognition in adaptive grippers for human–robot interaction. *Sensors*, 18 (3), 2018.

[6] L. Cao, R. Kotagiri, F. Sun, H. Li, W. Huang, and Z. M. M. Aye. Efficient spatio-temporal tactile object recognition with randomized tiling convolutional networks in a hierarchical fusion strategy. In *Thirtieth Aaai Conference on Artificial Intelligence*, 2016.

[7] Z. Pezzementi, E. Plaku, C. Reyda, and G. D. Hager. Tactile-object recognition from appearance information. *IEEE Transactions on Robotics*, 27(3):473–487, 2011.

[8] R. S. Dahiya, G. Metta, M. Valle, and G. Sandini. Tactile sensing—from humans to humanoids. *IEEE Transactions on Robotics*, 26(1):1–20, 2010.

[9] 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. *arXiv preprint arXiv:1902.06273*, 2019.

[10] Y. Li, J.-Y. Zhu, R. Tedrake, and A. Torralba. Connecting touch and vision via cross-modal prediction. *arXiv preprint arXiv:1906.06322*, 2019.

[11] S. Cai, K. Zhu, Y. Ban, and T. Narumi. Visual-tactile cross-modal data generation using residue-fusion GAN with feature-matching and perceptual losses. *IEEE Robotics and Automation Letters*, 6:7525–7532, 2021.

[12] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. *European Conference on Computer Vision (ECCV)*, 2020.

[13] W. Yuan, S. Dong, and E. H. Adelson. GelSight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12), 2017.

[14] M. Lambeta, P. Chou, S. Tian, B. H. Yang, B. Maloon, V. R. Most, D. Stroud, R. Santos, A. Byagowi, G. Kammerer, D. Jayaraman, and R. Calandra. DIGIT: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation. *arXiv preprint arXiv:2005.14679*, 2020.

[15] K. Patel, S. Iba, and N. Jamali. Deep tactile experience: Estimating tactile sensor output from depth sensor data. *IEEE International Conference on Intelligent Robots and Systems*, pages 9846–9853, 2020.

[16] S. Wang, M. Lambeta, P.-W. Chou, and R. Calandra. TACTO: A fast, flexible and open-source simulator for high-resolution vision-based tactile sensors. *arXiv preprint arXiv:2012.08456v1*, 2020.

[17] Z. Si and W. Yuan. Taxim: An example-based simulation model for gelsight tactile sensors. *arXiv preprint arXiv:2109.04027v1*, 2021.

[18] A. Agarwal, T. Man, and W. Yuan. Simulation of vision-based tactile sensors using physics based rendering. *arXiv preprint arXiv:2012.13184v3*, 2020.

[19] D. F. Gomes, P. Paoletti, and S. Luo. Generation of GelSight tactile images for sim2real learning. *arXiv preprint arXiv:2101.07169*, 2021.

[20] A. Padmanabha, F. Ebert, S. Tian, R. Calandra, C. Finn, and S. Levine. Omnitact: A multi-directional high resolution touch sensor. *CoRR*, abs/2003.06965, 2020.

[21] W. Chen, Y. Xu, Z. Chen, P. Zeng, R. Dang, R. Chen, and J. Xu. Bidirectional sim-to-real transfer for GelSight tactile sensors with CycleGAN. *IEEE Robotics and Automation Letters*, 7(3):6187–6194, 2022.

[22] T. Jianu, D. Gomes, and S. Luo. Reducing tactile sim2real domain gaps via deep texture generation networks. *International Conference on Robotics and Automation (ICRA)*, 2022.

[23] M. Mirza and S. Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014.

[24] R. Gao, Y.-Y. Chang, S. Mall, L. Fei-Fei, and J. Wu. ObjectFolder: A dataset of objects with implicit visual, auditory, and tactile representations. *arXiv preprint arXiv:2109.07991*, 2021.

[25] R. Gao, Z. Si, Y.-Y. Chang, S. Clarke, J. Bohg, L. Fei-Fei, W. Yuan, and J. Wu. ObjectFolder 2.0: A multisensory object dataset for sim2real transfer. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[26] N. Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995.

[27] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, 2016.

[28] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[29] 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*, 36(3):261–268, 2017.

[30] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4.

[31] T. Müller, A. Evans, C. Schied, and A. Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (SIGGRAPH)*, 41(4):102:1–102:15, 2022.