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

Shaohong Zhong, Alessandro Albini, Oiwi Parker Jones, Perla Maiolino, Ingmar Posner
Oxford Robotics Institute
University of Oxford, United Kingdom
{shaohong,alessandro,oiwi,perla,ingmar}@robots.ox.ac.uk

Abstract: Tactile perception is key for robotics applications such as manipulation. However, tactile data collection is time-consuming, especially when compared to vision. This limits the use of the tactile modality in machine learning solutions in robotics. In this paper, we propose a generative model to simulate realistic tactile sensory data for use in downstream tasks. Starting with easily-obtained camera images, we train Neural Radiance Fields (NeRF) for objects of interest. We then use NeRF-rendered RGB-D images as inputs to a conditional Generative Adversarial Network model (cGAN) to generate tactile images from desired orientations. We evaluate the generated data quantitatively using the Structural Similarity Index and Mean Squared Error metrics, and also using a tactile classification task both in simulation and in the real world. Results show that by augmenting a manually collected dataset, the generated data is able to increase classification accuracy by around 10%. 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. picking 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 is also critical for robotics applications such as manipulation and control [3] and object or texture recognition [4]. These tasks, especially those 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 the robot needs to physically interact with the environment and the object. While cameras can capture the global shape of an object, tactile sensors can only capture local features, and a long and time-consuming exploration procedure is usually required to capture the whole shape [7]. Beyond the problem relating to tactile exploration, tactile sensing is also still lacking standards at the hardware level [8]. For the same physical stimulus, the output of different tactile systems can differ significantly, thus limiting the validity of the collected data to a specific sensing technology.

Given the difficulties of tactile data collection, the problem of generating synthetic tactile sensor responses from data acquired using different modalities (which are easier to collect) becomes 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 input, these approaches can generate the corresponding tactile sensor output. One limitation of these vision-based generative approaches is that they 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 on the structure of the scene that we leverage for generating tactile images for 3D objects.

6th Conference on Robot Learning (CoRL 2022), Auckland, New Zealand.

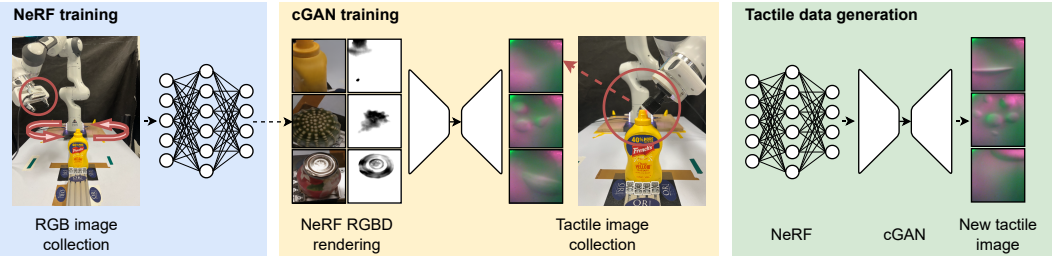

Figure 1: Overview of the framework. We first capture camera images around the object to train a NeRF model. We then collect a tactile image dataset to train a conditional GAN model that takes NeRF-rendered RGB-D images as input and outputs tactile images. Once the cGAN model is trained, we can query 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.

The main 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 an RGB-D camera to manually obtain a new input image each time a new object reading is desired. In addition, our framework is capable of transferring to a new sensor with a small fine-tuning dataset through the use of a deep generative model conditioned on a sensor background image. In this way, we can overcome the domain gap between different tactile sensors, and open up the potential to leverage different tactile datasets for even better 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*. One class of 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 [15, 16, 17, 18]. For example, the TACTO simulator presented in [15] allows one to emulate both Digit and the OmniTact sensors by simulating contacts in an off-the-shelf physics engine [14, 19]. Different simulators, such as Taxim aim to simulate the responses of a GelSight sensor by using a polynomial look-up table [16]. Another approach is to leverage methods such as physics-based rendering [17] and depth-maps from physics simulators [18]. Additional works attempt to address the sim2real issue in these tactile simulators through the use of CyclyGANs [20] and texture generation networks [21]. 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 [15, 16]. An accurate object model is also needed for simulating contacts [16]. In contrast, our proposed learning-based framework removes the need for modelling the sensor or the object.

A separate class of work explores learning-based approaches. A majority addresses the problem of generating tactile data from vision, using camera-based sensors, as both modalities encode information using the same data structure [10, 22, 9, 11]. For example, Li et al. [10] estimate the response of a GelSight sensor using a cGAN model [23] conditioned on a vision sequence that 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. [22] 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 viewpoints. Then, 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.

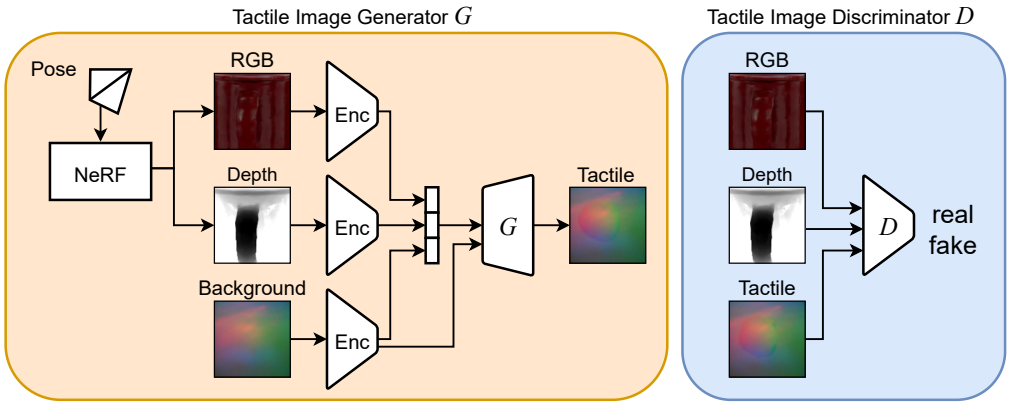

Figure 2: Proposed cGAN architecture. Given a desired pose, an RGB-D image is rendered from the NeRF model. The RGB, depth, and reference sensor background images are then encoded through separate encoders. The outputs of these encoders are concatenated before being fed to the generator network to generate the desired tactile image. During training, a classifier is used to differentiate whether a tactile 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.

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 to be 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 the RGB image, the depth image, and a reference background image $\mathbf{b}$ using separate encoders, 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 of 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 downstream robotics tasks. To collect tactile images, we employ Digit tactile sensors [14]. These sensors have two main components: an RGB camera and a compliant layer made of a soft gel. As a working principle, the camera captures the deformations of the soft gel caused by contact forces. For the simulation experiments, we use TACTO to simulate the Digit tactile readings [15]. We note that the TACTO simulator uses PyBullet [29] as its physics engine and PyRender [30] as the rendering engine.

4.1 Dataset generation

Simulation For simulation experiments, we select 27 common household objects from the scanned YCB dataset [31]. To train a NeRF model for each object, we collect 48 images taken from evenly sampled poses around a sphere centred 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, with the objects 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 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 9 common household objects from both the YCB dataset and other objects belonging to similar categories [31]. 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 them with the tactile images for training the cGAN model. For testing, we hold out 3 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. For each object, collecting images for the NeRF models takes approximately 20 minutes, whereas the collection of tactile data takes over 2 hours. The details of the objects are given in the Appendix.

All the objects in the simulated and real-world datasets are assumed to be rigid. For soft objects, their deformation when subjected to an external force, which is challenging to model, needs to be taken into account. By considering rigid objects only, the tactile sensor response mainly depends on the deformation of the soft layer of the sensor and on the contact force applied.

4.2 Evaluation details

As losses in cGAN are not directly indicative of the quality of the training results, it is difficult to use them to evaluate the realism of the generated tactile images. It is also difficult to manually evaluate the realism of such tactile images. Thus, we employ a set of approximate metrics and an example 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. The SSIM is a reference metric with a range of $[0, 1]$, whereas the MSE measures the difference in pixel values with a range of $[0, 65025]$, since we consider 8-bit images [32]. Next, we use tactile classification as an example task to evaluate the usefulness of the generated tactile datasets. The same evaluation pipelines are applied to 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 training datasets, one consisting of only ground-truth tactile images, the other consisting of both ground-truth and generated images. The test dataset consists of only ground-truth images. This is performed for both simulation and real-world data.

We also implement two benchmarks for comparison: 1) conditional GAN network for tactile image generation proposed by Lee et al. [9] 2) CycleGAN network [33]. In our experiments, both networks take RGB images as inputs and output tactile images at the corresponding pose. We use the default configurations proposed by the authors for our experiments.

5 Results and Discussion

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

5.1 Simulation results

From Table 1, we can see that our approach is able to 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 shown in Figure 3. On the novel view dataset, our approach is able to generate tactile images very similar to ground-truth ones, without having to explicitly model the object or the tactile sensor. The higher MSE value 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, as seen from Example 1 and 3 difference images in Figure 3, which indicate position mismatch.

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

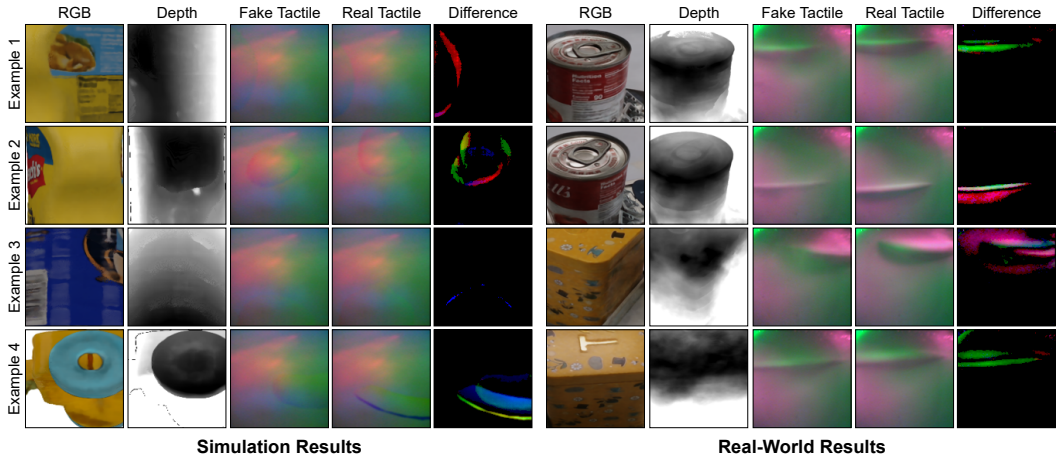

Figure 3: Qualitative results on novel object test sets. Our approach generates faithful tactile images in both simulated (left) and real-world (right) domains on objects it has not seen before in training. The columns show RGB images, depth images, fake tactile images, real tactile images, and the difference images between the real and the fake images. The difference images are thresholded to remove noise. The differences are mostly due to a mismatch in position or amount of indentation and exist mostly around the edges of the indentation. The fake tactile images are model generated; the real tactile images provide ground truth.

Test set	Method	SSIM ↑	MSE↓
Novel view	T-N	0.980 ± 0.001	14.5 ± 0.8
	Lee	0.678 ± 0.002	1040 ± 10
	Cycle	0.649 ± 0.002	1660 ± 10
Novel object	T-N	0.961 ± 0.000	84 ± 3
	Lee	0.687 ± 0.001	990 ± 10
	Cycle	0.660 ± 0.002	1610 ± 10
BG	-	0.921 ± 0.000	67.4 ± 3

(a) SSIM values between generated tactile images and ground-truth (simulated) tactile images.

Dataset	Accuracy/% ↑
Sim	85 ± 3
Sim + T-N	96 ± 2
Sim + Lee	86 ± 4
Sim + Cycle	80 ± 0

(b) Classification results for simulated dataset.

Table 1: Evaluation results for generated tactile images in simulation. Results shown are average values with standard errors. T-N: Touch NeRF (Ours). BG: Reference sensor background image.

leveraging NeRF models, we are also able to render additional RGB-D images of the object from novel view angles and generate additional tactile images to further augment the dataset at a low cost. Compared to the benchmarks, our approach outperforms by a large margin in both quantitative metrics and the classification task. This is likely due to the conditioning on depth, which enables our approach to predict tactile responses on 3D objects.

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 [19]. We collect simulated OmniTact images using a similar procedure as described in Section 4.1 and pair the tactile image with the corresponding RGB-D image rendered from NeRF models. 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 the 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 described in Section 4.2.

Using a much smaller fine-tuning dataset, the generated OmniTact tactile images still resemble the ground-truth (simulated) images, as shown in the SSIM and MSE results in Table 2a, and qualita-

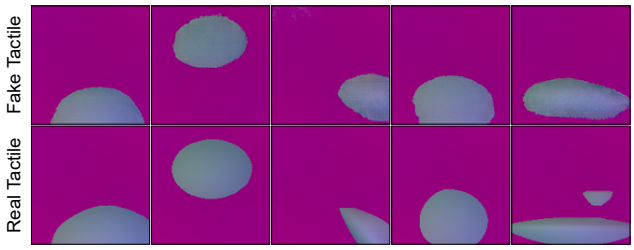

Figure 4: Example of generated tactile images for the simulated OmniTact sensor. In this case, the real tactile images are generated by the simulator.

Test set	Method	SSIM ↑	MSE↓
Novel view	T-N	0.865 ± 0.002	550 ± 10
	FS	0.829 ± 0.002	950 ± 20
Novel object	T-N	0.839 ± 0.003	760 ± 30
	T-N+	0.847 ± 0.003	760 ± 30
	FS	0.811 ± 0.005	1300 ± 60
BG	-	0.856 ± 0.001	1390 ± 20

(a) SSIM values between generated tactile images and ground-truth (simulated) tactile images.

Dataset	Accuracy/% ↑
Sim	86 ± 0
Sim + T-N	94 ± 1
Sim + FS	82 ± 1

(b) Classification results for using fine-tuning dataset.

Table 2: Evaluation results for generated tactile images for a new sensor in simulation. Results shown are average values with standard errors. T-N: Touch NeRF (Ours) with fine-tuning. T-N+: Touch NeRF with additional fine-tuning data. BG: Reference sensor background image. FS: Touch NeRF model trained from scratch.

tively in Figure 4. The lower SSIM value for novel objects is likely due to the presence of artefacts in the generated images. This could be remedied by adding an additional amount of fine-tuning data, as seen in the T-N+ results in Table2a, which include an additional 45 training images ($\sim$1/3 of the size of the original fine-tuning dataset). 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 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. This offers a potential avenue for further improving the data efficiency in tactile data generation using our approach.

5.2 Real-world results

From Figure 3 it can be seen that the proposed framework is able to generate realistic tactile images in the real world 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 soft gel. 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 require the force to be within a range during tactile data collection.

Table 3 shows that the generated tactile images are able to capture the key properties of ground-truth tactile images collected in the real world. The SSIM values are lower than those in simulation in Table 1, which is expected due to the presence of noise in the real world. Additionally, classification results in Table 3b indicate that the generated tactile images are useful in increasing the accuracy of the downstream tactile classification task. The failure modes exhibited in the classification experiments are found to be mostly due to the similar tactile features exhibited in some test objects.

It can be seen that our approach outperforms both benchmarks in quantitative metrics and the classification task. It should be noted that the RGB-D images rendered by real-world NeRF models

contain more artefacts than those in simulation, as seen in Figure 3. However, even using the noisier RGB-D renderings, our approach is still able to generate high-quality tactile images for 3D objects. Additionally, our approach offers more generalisability on two fronts: firstly, as seen from the results in Table 3a, our approach outperforms in the novel object test set by a large margin; secondly, by leveraging a NeRF model, our approach is able to generate high-quality tactile images from novel viewpoints. This further removes the need to manually retake images for every new tactile image, as required by the baseline approaches.

Test set	Method	SSIM ↑	MSE↓
Novel view	T-N	0.887 ± 0.001	56 ± 1
	Lee	0.708 ± 0.006	960 ± 50
	Cycle	0.781 ± 0.006	420 ± 60
Novel object	T-N	0.838 ± 0.001	225 ± 3
	Lee	0.723 ± 0.003	850 ± 30
	Cycle	0.765 ± 0.002	390 ± 10
BG	-	0.814 ± 0.000	239 ± 3

(a) SSIM values between generated tactile images and ground-truth (real) tactile images.

Dataset	Accuracy/% ↑
Real	74 ± 1
Real + T-N	83 ± 2
Real + Lee	76 ± 2
Real+ Cycle	70 ± 0
T-N*	70 ± 6

(b) Classification results for real-world dataset.

Table 3: Evaluation results for generated tactile images in the real world. Results shown are average values with standard errors. T-N: Touch NeRF (Ours). BG: Reference sensor background image. T-N*: Touch NeRF deployed 'in the wild' (training dataset contains no real data).

To illustrate the proposed approach in deployment, we collect RGB image data (48 images per object) for three additional objects on a different background using a mobile phone and trained corresponding NeRF models (with a lower resolution setting for faster training). We then generate tactile images for each object using our trained real-world cGAN model, and evaluate the quality of the generated tactile images in a similar classification task. The classifier is trained only on generated data, and is tested on manually-collected real tactile data. The T-N* result in Table 3b shows that the proposed approach is still able to achieve high accuracy in this more challenging setting. In this case, the data collection took around 2 minutes per object with casually taken camera images, further demonstrating the ease of deployment of our approach and the potential for reducing human efforts in tactile data collection.

6 Conclusion and Limitations

In this paper, we present a novel framework for the generation of tactile data for camera-based tactile sensors. The framework leverages NeRF models to render RGB-D images of an object from desired poses, and passes the RGB-D images to a cGAN model to generate the desired tactile images. Compared to state-of-the-art, this approach allows for the generation of tactile images for 3D objects from arbitrary viewpoints, without the need for accurate sensor or object models. Results demonstrate that the generated tactile images are structurally similar to ground-truth images, and are useful in downstream robotics tasks such as tactile classification. We further demonstrate the potential of a novel capability to transfer from one tactile modality to another with a small fine-tuning dataset.

One potential limit of the approach is that a NeRF model needs to be trained for each object. However, it should be noted that the collection of image data for training NeRF models requires much less human effort than the collection of tactile data, as illustrated by our deployment experiment. This is also arguably simpler than building a scanned object model. Additionally, recent advances in accelerating NeRF training also significantly mitigate the computation time needed [34].

We also assumed to operate only on rigid objects, with a fixed force range and orientation (pitch and yaw) of the tactile sensor. Soft objects might require modelling the non-linear deformation of the object when in contact, which adds additional complexity in predicting the tactile response. This is currently out of the scope of the paper. It must also be noted that even under the rigid object assumption, generating tactile images for 3D objects is challenging and is still an open problem for the tactile-sensing community. The generalisation to different forces and orientations, and potentially soft objects, is left to future work.

Acknowledgments

We thank Jun Yamada for help with the experimental set-up, and Jack Collins and Alexander Mitchell for suggestions on the paper draft, and Yizhe Wu and other members of the Applied Artificial Intelligence Lab and Soft Robotics Lab for helpful discussions. We thank the anonymous reviewers for their valuable feedback in revising the paper. This work was supported in part by the UKRI/EPSRC Programme under Grant EP/V000748/1, and by a CSC-PAG Oxford Scholarship. We would like to acknowledge the use of the University of Oxford Advanced Research Computing (ARC) facility in carrying out this work. http://dx.doi.org/10.5281/zenodo.22558. We would also like to acknowledge the use of the SCAN facility.

References

[1] 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*, 3(4):3300–3307, 2018. doi:10.1109/LRA.2018.2852779.

[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. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6444–6451, 2021. doi:10.1109/ICRA48506.2021.9561969.

[3] Z. Kappassov, J.-A. Corrales, and V. Perdereau. Tactile sensing in dexterous robot hands – review. *Robotics and Autonomous Systems*, 74:195–220, 2015. ISSN 0921-8890. doi:https://doi.org/10.1016/j.robot.2015.07.015. URL https://www.sciencedirect.com/science/article/pii/S0921889015001621.

[4] S. Luo, J. Bimbo, R. Dahiya, and H. Liu. Robotic tactile perception of object properties: A review. *Mechatronics*, 48:54–67, 2017. ISSN 0957-4158. doi:https://doi.org/10.1016/j.mechatronics.2017.11.002. URL https://www.sciencedirect.com/science/article/pii/S0957415817301575.

[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. ISSN 1424-8220. doi:10.3390/s18030692. URL https://www.mdpi.com/1424-8220/18/3/692.

[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. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1), Mar. 2016. doi:10.1609/aaai.v30i1.10412. URL https://ojs.aaai.org/index.php/AAAI/article/view/10412.

[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. doi:10.1109/TRO.2011.2125350.

[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. doi:10.1109/TRO.2009.2033627.

[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. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 4276–4282, 2019. doi:10.1109/ICRA.2019.8793763.

[10] Y. Li, J.-Y. Zhu, R. Tedrake, and A. Torralba. Connecting touch and vision via cross-modal prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10609–10618, 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(4):7525–7532, 2021. doi:10.1109/LRA.2021.3095925.

[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. *Commun. ACM*, 65(1): 99–106, dec 2021. ISSN 0001-0782. doi:10.1145/3503250. URL https://doi.org/10.1145/3503250.

[13] W. Yuan, S. Dong, and E. H. Adelson. Gelsight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12), 2017. ISSN 1424-8220. doi:10.3390/s17122762. URL https://www.mdpi.com/1424-8220/17/12/2762.

[14] M. Lambeta, P.-W. Chou, S. Tian, B. 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. *IEEE Robotics and Automation Letters*, 5(3):3838–3845, 2020. doi:10.1109/LRA.2020.2977257.

[15] 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. *IEEE Robotics and Automation Letters*, 7(2):3930–3937, 2022. doi:10.1109/LRA.2022.3146945.

[16] Z. Si and W. Yuan. Taxim: An example-based simulation model for gelsight tactile sensors. *IEEE Robotics and Automation Letters*, 7(2):2361–2368, 2022. doi:10.1109/LRA.2022.3142412.

[17] A. Agarwal, T. Man, and W. Yuan. Simulation of vision-based tactile sensors using physics based rendering. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–7, 2021. doi:10.1109/ICRA48506.2021.9561122.

[18] D. F. Gomes, P. Paoletti, and S. Luo. Generation of gelsight tactile images for sim2real learning. *IEEE Robotics and Automation Letters*, 6(2):4177–4184, 2021. doi:10.1109/LRA.2021.3063925.

[19] A. Padmanabha, F. Ebert, S. Tian, R. Calandra, C. Finn, and S. Levine. Omnitact: A multi-directional high-resolution touch sensor. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 618–624, 2020. doi:10.1109/ICRA40945.2020.9196712.

[20] 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. doi:10.1109/LRA.2022.3167064.

[21] T. Jianu, D. F. Gomes, and S. Luo. Reducing tactile sim2real domain gaps via deep texture generation networks. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 8305–8311, 2022. doi:10.1109/ICRA46639.2022.9811801.

[22] K. Patel, S. Iba, and N. Jamali. Deep tactile experience: Estimating tactile sensor output from depth sensor data. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9846–9853, 2020. doi:10.1109/IROS45743.2020.9341596.

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

[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. In *5th Annual Conference on Robot Learning*, 2021. URL https://openreview.net/forum?id=A9P78VBwYKM.

[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. doi:10.1109/2945.468400.

[27] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, 2017. doi:10.1109/CVPR.2017.632.

[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. URL http://jmlr.org/papers/v15/srivastava14a.html.

[29] E. Coumans and Y. Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. 2016.

[30] M. Matl. Pyrender. https://github.com/mmatl/pyrender, 2019.

[31] 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. doi:10.1177/0278364917700714. URL https://doi.org/10.1177/0278364917700714.

[32] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi:10.1109/TIP.2003.819861.

[33] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251, 2017. doi:10.1109/ICCV.2017.244.

[34] T. Müller, A. Evans, C. Schied, and A. Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022. doi:10.1145/3528223.3530127. URL https://doi.org/10.1145/3528223.3530127.

[35] L. Yen-Chen. Nerf-pytorch. https://github.com/yenchenlin/nerf-pytorch/, 2020.

[36] 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. doi:10.1007/978-3-319-24574-4_28.