IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

Marker-Embedded Tactile Image Generation via Generative Adversarial Networks

Won Dong Kim, Sanghoon Yang, Woojong Kim,
Jeong-Jung Kim, Chang-Hyun Kim, and Jung Kim, *Member, IEEE*

***Abstract*—Data-driven methods have been successfully applied to images from vision-based tactile sensors to fulfill various manipulation tasks. Nevertheless, these methods remain inefficient because of the lack of methods for simulating the sensors. Relevant research on simulating vision-based tactile sensors generally focus on generating images without markers, owing to the challenges in accurately generating marker motions caused by elastomer deformation. This disallows access to tactile information deducible from markers. In this work, we propose a generative adversarial network (GAN)-based method to generate realistic marker-embedded tactile images in GelSight-like vision-based tactile sensors. We trained the proposed GAN model with an aligned real tactile and simulated depth image dataset obtained from deforming the sensor against various objects. This allows the model to translate simulated depth image sequences into RGB tactile images with markers. Furthermore, the generator in the proposed GAN allows the network to integrate the history of deformations from the depth image sequences to generate realistic marker motions during the normal and lateral sensor deformations. We evaluated and compared the positional accuracy of the markers and image similarity metrics of the images generated via our method with those from prior methods. The generated tactile images from the proposed model show a 28.3 % decrease in marker positional error and a 93.5 % decrease in the image similarity metric (MSE) compared to those generated by previous methods, validating the effectiveness of our approach. The materials used are available at https://github.com/WonDKim/marker-simulation.**

***Index Terms*—Force and Tactile Sensing, Deep Learning Methods, Simulation and Animation**

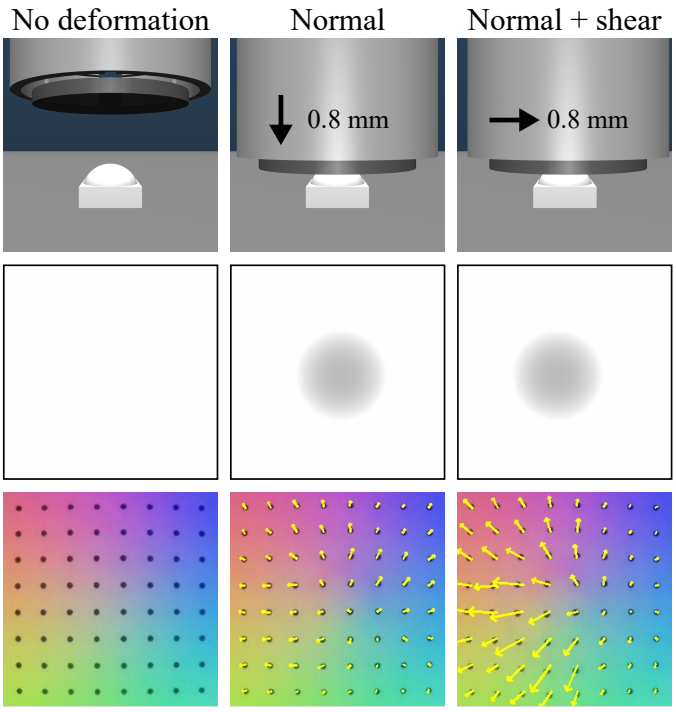

Fig. 1. Our generator network generates realistic marker-embedded tactile images (third row) from simulated depth image sequences (second row) obtained from the simulated tactile sensor (first row). The marker motion vectors are scaled up by 5 for clearer visualization.

I. Introduction

TACTILE sensing is essential for robots to interact physically with objects and the environment [1]. Robots use tactile sensors to sense external physical stimuli resulting from contact and perceive information about the physical properties of objects [2]. As this information is not readily deducible from vision sensors, tactile sensors are key complements to vision sensors in performing contact-rich manipulation tasks [3].

Recent advances in various types of vision-based tactile sensors (e.g., GelSight and its variants [4]–[7], TacTip family [8], Soft-Bubble [9], and UVtac [10]) have enabled high-resolution tactile sensing through a relatively simple and inexpensive fabrication process. Vision-based tactile sensors transduce contacts by capturing images of the elastomer deformation using their embedded camera. Particularly, GelSight and its variants use the photometric stereo method [11] to reconstruct heightmaps using RGB intensity changes in the images. Markers are generally printed on the surface of the elastomer and move as the surface deforms, providing additional information about tactile events. Features including contact geometry measurement and slip detection [12] and force estimation [13] can be extracted from these images.

Data-driven methods have been applied to GelSight tactile images for perception and control tasks [14]–[16]. Similar to data-driven methods in other robotic applications [17], [18], the performance and efficiency of data-driven methods

Manuscript received: December, 8, 2022; Revised April, 11, 2023; Accepted May, 23, 2023. This paper was recommended for publication by Editor A. Banerjee upon evaluation of the Associate Editor and Reviewers' comments. This study is a part of the research project, "Development of core machinery technologies for autonomous operation and manufacturing (NK242H)", which has been supported by a grant from National Research Council of Science & Technology under the R&D Program of Ministry of Science, ICT and Future Planning. *(Corresponding author: Jung Kim.)*

W. D. Kim, S. Yang, W. Kim, and J. Kim are with the Department of Mechanical Engineering at KAIST, Daejeon, 34141, Republic of Korea (e-mail: kwd92@kaist.ac.kr; ysh2146@kaist.ac.kr; kwjong2028@kaist.ac.kr; jungkim@kaist.ac.kr)

J-J. Kim and C-H. Kim are with the Korea Institute of Machinery and Materials, Daejeon, 34103, Republic of Korea (e-mail: rightcore@kimm.re.kr; chkim78@kimm.re.kr)

Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

in vision-based tactile sensor applications can be increased through simulations. Nevertheless, simulations for these sensors are limited owing to challenges in simulating both the nonlinear mechanical response of the elastomer and the complex optical response due to the internal illumination conditions and elastomer deformations.

Several studies have been conducted to solve these problems [19]–[21]. These studies mainly on the optical simulation of vision-based tactile sensors (i.e., using tactile images without markers). However, tactile images with markers are used in many contact-rich manipulation tasks [14], [15], [22] as information on contact forces can be determined by examining marker motions. Thus, simulating tactile images with markers should be treated with equal importance as the realistic simulation of the reflective layer. Si and Yuan [23] proposed a simulation model that implements a marker motion field-simulating method based on an approximated finite element method (FEM), independent of its optical simulating method.

In this work, we propose a method for generating tactile images with markers for vision-based tactile sensors like GelSight. We approach the problem as an image-to-image translation (I2I) task, wherein the objective is to translate depth images generated by the depth camera in the simulated sensor into tactile images captured by the RGB camera in the real sensor. We constructed a generative adversarial network (GAN) framework [24] with a 3D-UNet architecture with LSTM bottleneck to solve this problem. The proposed model allows the generator to distinguish deformations with different deformation histories by referring to the preceding information in the simulated depth image sequence input (Fig. 1). The GAN model was trained with real tactile and simulated depth image data pairs collected through a self-made GelSight-like sensor. Unlike the method in [23], where a separate FEM computation is required for marker generation, our method directly generates a complete marker-embedded tactile image from simulated depth image sequences using a single pipeline (Fig. 2), owing to the fact that the generator learns both the optical response and the marker motions caused by elastomer deformation. We quantitatively evaluated the quality of the generated tactile images in two aspects. First, we compared errors in marker positions and marker motion vectors of the generated images to those of other methods. Second, we evaluated and compared the image similarity metrics of the marker-excluded reflective layer regions of the generated images. Our evaluation results show that the proposed method generates marker-embedded tactile images with marker position errors and mean squared error (MSE) image similarity metrics that are, respectively, 28.3 % lower than the illumination model-based method and 93.5 % lower than the FEM-based method. These results validate the use of our approach to generating realistic marker-embedded tactile images.

II. Related Work

A. Simulation of Vision-based Tactile Sensor

Studies have emphasized the need to create a simulation framework for vision-based tactile sensors to overcome limitations in applying data-driven methods using real sensors.

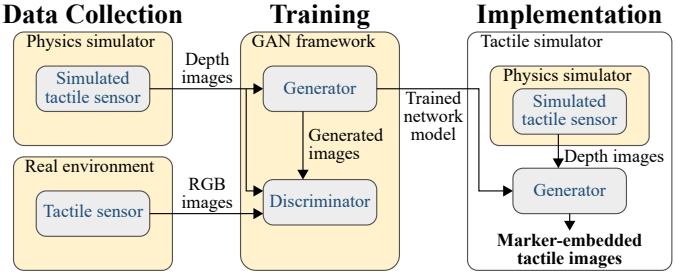

Fig. 2. Building process (data collection and training) and pipeline of the final implemented GAN-based tactile image generation method.

The work from Gomes *et al.* [19] was one of the first research to address this need. The authors proposed using a simulated depth camera in the Gazebo simulator to capture depth images of contact objects. Simulated depth images were processed using a difference of Gaussian filter to smoothen the contact edges. Phong's rendering model was then applied based on the arrangement of the LEDs to approximate the light reflection on the deformed reflective layer. Agarwal *et al.* [21] developed a simulation framework using physics-based rendering techniques. Wang *et al.* introduced TACTO [20], an open-source simulation framework built on the PyBullet physics and Pyrender rendering engine. TACTO was designed to be extendable to other variations of GelSight (e.g., OmniTact [6] and DIGIT [7]).

The mentioned studies have only focused on simulating and generating vision-based tactile sensor images with reflective layers without printed markers. Si and Yuan [23] were the first to propose a simulation framework that considered both the reflective layer and motion of the printed markers. Their framework comprised two independent pipelines: an example-based pipeline to simulate the reflective layer and an approximated FEM-based pipeline to simulate the marker motion field. The pipeline for simulating the marker motion field used the precomputed quasi-static linear elastic model [25]. Displacements of the surface nodes were calculated using precomputed linear tensors between the nodes of the elastomer surface mesh. Node displacements were interpolated to create motion field maps in x, y, and z directions, which were used to generate markers by locating their positions within the motion field maps. Our method naturally integrates the reflective layer and marker simulation pipelines into a single data-driven pipeline. This eliminates the need for any optical and mechanical modeling, which requires arduous tuning of the light source and elastic model parameters and is susceptible to errors.

B. GAN for Image-to-Image Translation

GANs [24] have been widely used to solve I2I problems wherein the main objective is to transfer images from a source domain to a different domain while preserving the core contents of the original images [26]. In robotics, GANs for I2I have been primarily applied to systems using vision sensors. These frameworks have been used successfully for domain adaptation [27], [28] and data augmentation [29], [30] applications. Pang *et al.* [26] presented a thorough review of

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

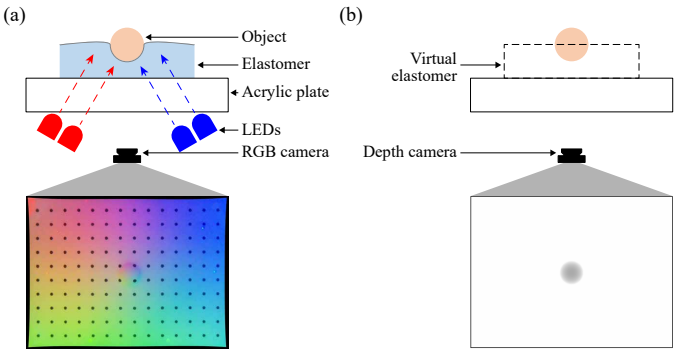

Fig. 3. Simplified sectional view of (a) real and (b) simulated tactile sensor. The former outputs an RGB tactile image, and the latter outputs a depth image.

the different types, characteristics, and applications of GANs for I2I.

Some research has leveraged the capability of GAN in I2I for generating tactile images. Jianu *et al.* [31] used GAN to generate textures on tactile images using cycle consistency loss to avoid needing to collect aligned pairs of real and simulated images. Moreover, masks were applied to the contacted regions to restrict the texture generation to the areas in contact with the objects. Chen *et al.* [32] used CycleGAN [33] to generate realistic textured tactile images from simulated images obtained from Phong's model and vice versa. In both works, the simulated depth images are processed using illumination models, which require careful adjustments of the light source parameters and do not solve the problem of generating tactile images with markers. Our method eliminates the need to use illumination models by directly transferring simulated depth images to RGB tactile images and provides a solution for generating marker-embedded tactile images.

III. Methods

We built and applied the marker-embedded tactile image generator through data collection, training, and final implementation (Fig. 2). First, we collected aligned simulated depth and real tactile image datasets. Details of the simulated counterpart of a vision-based tactile sensor and its image outputs are explained in Section III-A. We then used the image dataset to train the generator and discriminator in the GAN framework. Section III-B provides a thorough description of the network architectures in the GAN. Finally, we integrate the trained generator with the simulator to generate realistic tactile images from simulated depth images directly.

A. Vision-Based Tactile Sensor in a Simulator

As reproducing the behavior of the elastomer in simulators is difficult, we use an approach similar to that in [19], where the RGB camera (Fig. 3 (a)) is replaced with a depth camera (Fig. 3 (b)). We replace the elastomer with a virtual elastomer, which is a transparent solid with low contact stiffness. To ensure that the depth image captured by the depth camera contains objects only within the virtual elastomer, the raw depth image D is thresholded to obtain the final depth image D' (Fig. 3 (b)):

$$D'(x,y) = \begin{cases} D(x,y) & \text{if } D(x,y) \leq D_{max} \\ D_{max} & \text{otherwise} \end{cases} \quad (1)$$

Unlike in [19], where the depth image is further processed with filters and overlaid with reflected RGB surfaces computed using different illumination models, our method has no additional depth image processing steps. This eliminates the need for a meticulous placement of LEDs in the simulated sensor and complex calculations for applying illumination models.

B. Proposed Generative Adversarial Network Model

Our GAN primarily aims to train a generator network to translate simulated depth images to realistic RGB tactile images with markers. Hence, it uses depth images as inputs to the generator and discriminator, while using the real tactile images as input to the discriminator (Fig. 4). We based our model on the Pix2Pix model [34], but added some modifications. Most notably, this model uses a sequence of depth images as input instead of a single image. The depth image sequence allows the generator to identify past deformation events rather than a single snapshot of the deformed elastomer.

1) Network Architecture: The architecture of the generator (G) takes the form of 3D-UNet, an encoder-decoder 3D convolutional neural network architecture. In this architecture, the encoder and decoder blocks are connected through skip connections. The proposed model replaces the bottleneck layer with two LSTM layers. Moreover, the architecture of the proposed model processes image sequences and refers to previous deformation information for more realistic image generation. The generator is fed a depth image sequence of size (n, 256, 256, 3). The value of n varies depending on the degree of deformation in the images, where the maximum deformation is set to normal and shear deformations of 1.5 mm and 1 mm, respectively ($n = 26$). The deformation rate is assumed to be uniform, and each image in the sequence corresponds to a deformation step of 0.1 mm. The input image sequence passes through the encoder, two LSTM layers with a hidden size of 512, and the decoder to produce an output of shape (n, 256, 256, 3), which corresponds to n RGB images of size of 256 $\times$ 256.

The architecture of the discriminator (D) follows the structure of PatchGAN [34]. However, each convolution layer is replaced with a 3D convolution layer. D receives an input tensor of shape (n, 256, 256, 6), which results from concatenating depth image sequences with either the generated image sequence ($G(I_{sim})$) or real tactile image sequence (I_{real}). D outputs a tensor of size (n, 30, 30, 1), wherein each element ranges in [0, 1] owing to the final hyperbolic tangent ($tanh$) activation. Each element evaluates whether the corresponding patch of the image is derived from a real or a generated image. The model's capability to process and generate image sequences and provide discriminator gradient signals to all images in the sequence allows us to train the model with sequences corresponding to the maximum shear deformations, without having to train with sequences of intermediate deformations.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

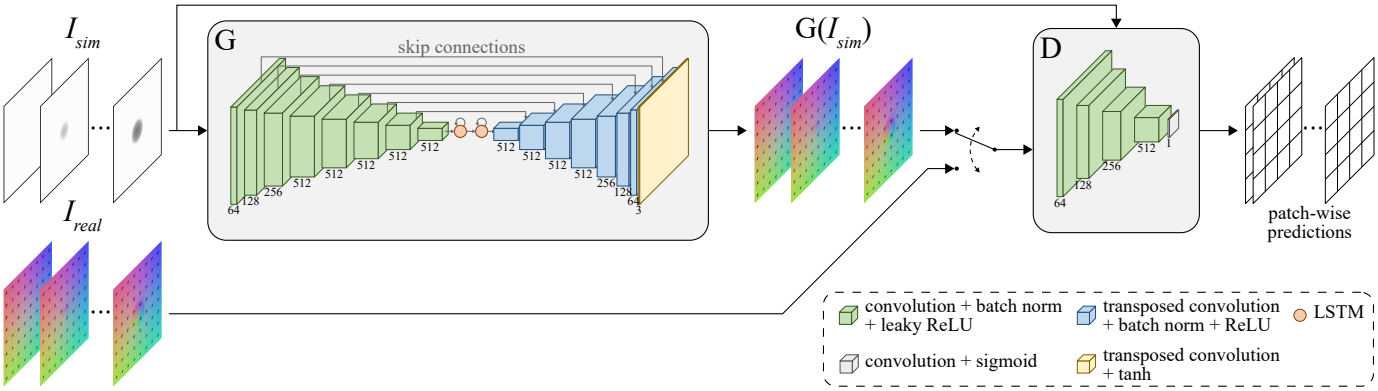

Fig. 4. GAN architecture for our proposed method. All convolution layers are 3D convolution layers. The number below the convolution layers corresponds to that of output channels.

2) Loss Function: Loss function $\mathcal{L}$ consists of the reconstruction and adversarial loss terms. First, reconstruction loss acts as a loss term to increase generated image quality by comparing them directly to real images. The reconstruction loss uses the pixel-wise L1 error as follows:

$$\mathcal{L}_{\mathrm{L1}}(G) = \|I_{real} - G(I_{sim})\|_1 \tag{2}$$

Adversarial loss is the loss term using the error in discriminating whether the input to the discriminator is real or fake. Adversarial loss can be expressed as follows:

$$\mathcal{L}_{adv}(G, D) = \log D + \log\left(1 - D(G(I_{sim}))\right) \tag{3}$$

Reconstruction and adversarial loss are then combined to make the overall loss function as follows:

$$\mathcal{L} = \mathcal{L}_{adv}(G, D) + \lambda \mathcal{L}_{\mathrm{L1}}(G), \tag{4}$$

where λ is the hyperparameter for setting the ratio of weights between $\mathcal{L}_{\mathrm{L1}}$ and $\mathcal{L}_{adv}$. The hyperparameter was set to $\lambda = 100$ for this work based on a self-investigation showing that the model trained with this value has the most balanced image generation performance across different types of deformations.

IV. Experimental Setup

A. Real Environment Setup

We fabricated a self-made vision-based tactile sensor (Fig. 5 (a)) to validate the proposed generative method. The structure of the sensor (Fig. 5 (b)) is similar to those using the photometric stereo method, consisting of a camera (Raspberry Pi Camera Module v2), LEDs (WS2812B, Worldsemi Co., Ltd.), and a clear elastomer skin (Solaris, Smooth-On Inc.) over a rigid transparent plate. We added a reflective layer and markers to the surface of the elastomer skin using the fabrication process presented in [5]. We designed the 3D-printed case to ensure the convenient assembly of the sensor components and secure mount onto the UR5e robot (Universal Robots, Denmark).

Fig. 5 (c) shows the setup for collecting real tactile images. In the real environment, the sensor was mounted onto the UR5e robot. We then placed different objects at predefined positions relative to the robot base. We integrated the control of the UR5e and the acquisition of the tactile image into a Robot Operating System (ROS) Python script. Using the ROS script, we could control the sensor pose and acquire tactile images simultaneously.

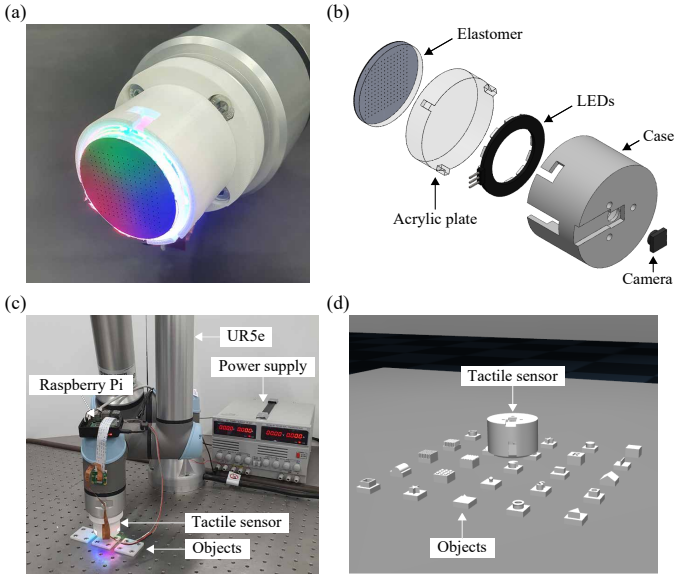

Fig. 5. (a) The photo and (b) schematic of the self-made sensor. (c) The real and (d) simulated experiment environment. In (c), the sensor is mounted onto the UR5e robot. In (d), we controlled the sensor pose using four unseen virtual actuators.

B. Simulated Environment Setup

We used the MuJoCo simulator [35] to construct the simulated environment (Fig. 5 (d)). In the simulation model, we replaced the elastomer of the sensor with a transparent structure. The embedded RGB camera is replaced with an RGB-D camera (Fig. 3 (b)). Unlike the real environment setup, we controlled the pose of the simulated sensor using three prismatic joints (x, y, and z axes) and a revolute joint (about the z axis) via the *mujoco-py* Python package.

In MuJoCo, the depth camera outputs values in depth buffers instead of the actual depth values. The true depth (z) is calculated from the depth buffer (z_b) as follows:

$$z = \frac{2 z_{far} z_{near}}{z_{far} + z_{near} - (2 z_b - 1)(z_{far} - z_{near})}, \tag{5}$$

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

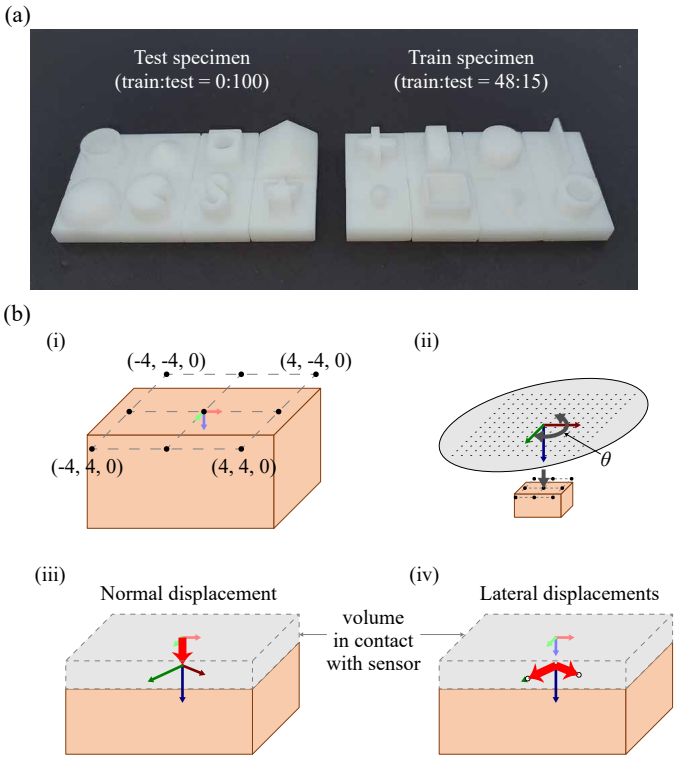

Fig. 6. (a) The 3D-printed objects set. Images from contacting the objects on the left (*circleshell*, *cone*, *cubehole*, *doubleslope*, *hemisphere*, *pacman*, *S*, and *star*) are used as test data. Most of the images from contacting the objects on the right (*cross*, *cuboid*, *cylinder*, *line*, *sphere*, *squareshell*, *tetrahedron*, and *torus*) are used as training data. (b) Illustration of the deformation process. (i) A total of nine points are selected as initial deformation points. (ii) The centroid of the elastomer is aligned with one of the nine points at an angle of θ. (iii) The elastomer is first displaced in the normal direction by δ_z, (iv) followed by a lateral displacement of either δ_x or δ_y along the sensor frame.

where z_{far} and z_{near} are the distance between the depth camera and the farthest view plane and that between the depth camera and the nearest view plane, respectively. After applying Equation (1) to the true depth image, this is converted to a grayscale image, where RGB codes (0, 0, 0) and (255, 255, 255) correspond to depth values of 0 and 3 mm (the height of the elastomer), respectively.

C. Dataset Collection

We collected aligned tactile and simulated depth images from the real and simulated environments to train and evaluate the proposed GAN. Fig. 6 (a) shows the 16 objects used to complete the dataset. The objects were printed using the stereolithography 3D printing technology. They were designed to fit into a convex hull with dimensions of $15 \times 15 \times 10$ mm^3.

To collect an aligned dataset of normally and laterally deformed real and simulated images, we followed the deformation process in Fig. 6 (b) in both real and simulated setups. First, we aligned the centroid of the elastomer to one of the nine initial points, where the points form a grid of 4 mm intervals centered at the centroid of the uppermost plane of the object. The initial sensor pose is varied between $[-90°, 90°]$ about the z axis at an interval of 30°. We can express the initial sensor pose (x, y, z, θ) relative to the object reference frame as follows:

$$x, y \in \{-4, 0, 4\}, \qquad z = 0,$$
$$\theta \in \left\{ \frac{\pi r}{6} \;\middle|\; -3 \leq r \leq 3, r \in \mathbb{Z} \right\}$$

Once the sensor is set to its initial pose, it is then displaced downwards by $\delta_z \in \{0.1\gamma \mid 0 < \gamma \leq 15, \gamma \in \mathbb{Z}\}$. This sensor is then displaced laterally along either its x or y axis up to 1 mm at 0.1 mm interval. The process is repeated for all δ_z up to 1.5 mm. We performed all deformation processes slowly (0.05 mm/s) with sufficient rest time between deformations to negate any time- or rate-dependent nonlinear effects. We captured tactile and simulated depth images at every normal and lateral interval, totaling 19845 tactile and depth image pairs per object. This is equivalent to a total image pair dataset size of 318k for 16 objects.

We split this dataset into train and test sets based on the contacted objects and initial contact pose. We classified all images collected by contacting the *circleshell*, *cone*, *cubehole*, *doubleslope*, *hemisphere*, *pacman*, *S*, and *star* as test set (Fig. 6 (a)). From the images collected by contacting the remaining objects, we randomly chose 15 initial poses from the possible 63 per object. We set the contact images starting from the chosen 15 initial poses as test data, allowing us to evaluate the intra-object generalization performance of the GAN. We used the remaining image pairs as train data.

D. Training Details

We used the Adam optimization [36] as the stochastic gradient descent method with an initial learning rate of $\alpha_0 = 0.0002$ and momentum parameters of $\beta_1 = 0.5$ and $\beta_2 = 0.999$. We trained the networks for 20 epochs, where the learning rate is kept at α_0 for the first 10 epochs and is linearly decayed to zero in the latter 10 epochs. For generalized performance evaluation, the models were trained five times with different initial weights. We trained the networks of all GAN models under the same conditions. We conducted all training and testing computations using an NVIDIA TITAN Xp GPU, where a single training and inference iteration takes on average 1193 and 12 ms, respectively.

V. Results

In this section, we evaluated and compared the quality of the images generated by the proposed model with that by other methods. Fig. 7 shows examples of generated tactile images. We combined the methods in [19] (Phong) and [23] (Taxim) to produce marker-embedded tactile images. Images generated from Phong's model seem unnatural owing to the thick and sharp contact edges. The Pix2Pix results show that the images contain artifacts and blurry markers regardless of the degree of deformation and contact objects. Overall, the images generated by the proposed and 3D-UNet are closest to the real images in appearance.

For a more quantitative analysis, we first evaluated the proposed model's marker generation capability and accuracy. Second, we evaluated the image similarity of the non-markered

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

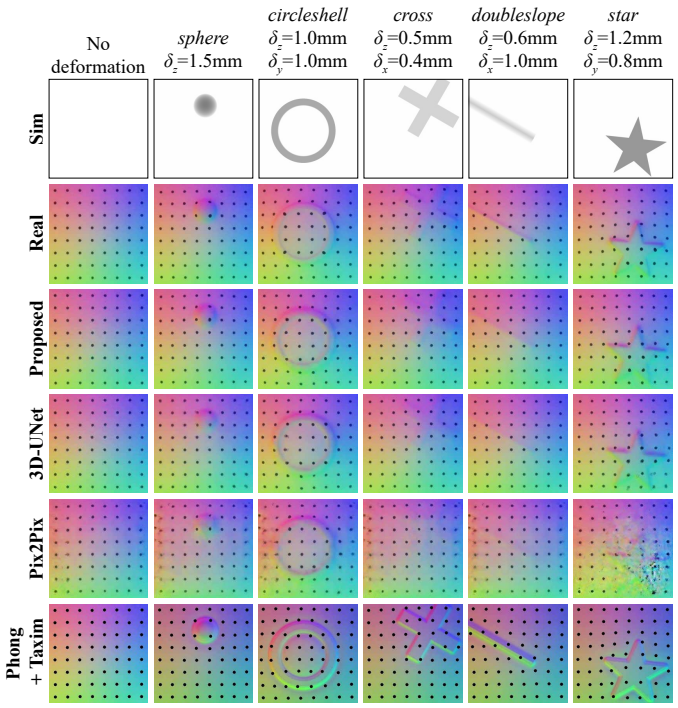

Fig. 7. Examples of simulated depth images (first row), real tactile images (second row), and generated tactile images (third to last row) for different deformations.

TABLE I
MEAN AND STANDARD DEVIATION (IN BRACKETS) OF POSITIONAL ERROR OF MARKERS, ERROR VECTOR MAGNITUDE (ϵ_{EVM}) OF MARKER MOTION VECTORS, AND PERCENTAGE OF DEGENERATE IMAGES

	d (px)	ϵ_{EVM} (px)	Degenerate images (%)
Taxim [23]	205	205	-
Pix2Pix	193 (28)	204 (12)	46.9 (25.5)
3D-UNet	169 (16)	174 (12)	3.78 (2.81)
Proposed	**147 (8)**	**154 (6)**	**0.522 (0.635)**

regions of the generated tactile images by the proposed model. In both analyses, the performances were compared to those of a variant of the proposed model, which does not have the LSTM layers in the generator (3D-UNet), and the standard Pix2Pix model. Additionally, we used the performances of Taxim and Phong as baselines in the marker generation and image similarity evaluations, respectively.

A. Marker Motion

To quantitatively evaluate the performance of different approaches in generating marker-embedded tactile images, we examined the positional error, error vector magnitude (ϵ_{EVM}) of the marker motion vectors, and percentage of generated images that could not reconstruct the markers fully. We computed the sum of Euclidean distances between real and generated marker positions for positional error d as follows:

$$d = \sum_{i=1}^{n} \sqrt{(x_{i_{gen}} - x_{i_{real}})^2 + (y_{i_{gen}} - y_{i_{real}})^2}, \quad (6)$$

where x_i and y_i are the horizontal and vertical pixel-wise centroid position of the i^{th} marker, respectively, and n is the number of markers, equal to 64 in this work. We calculated

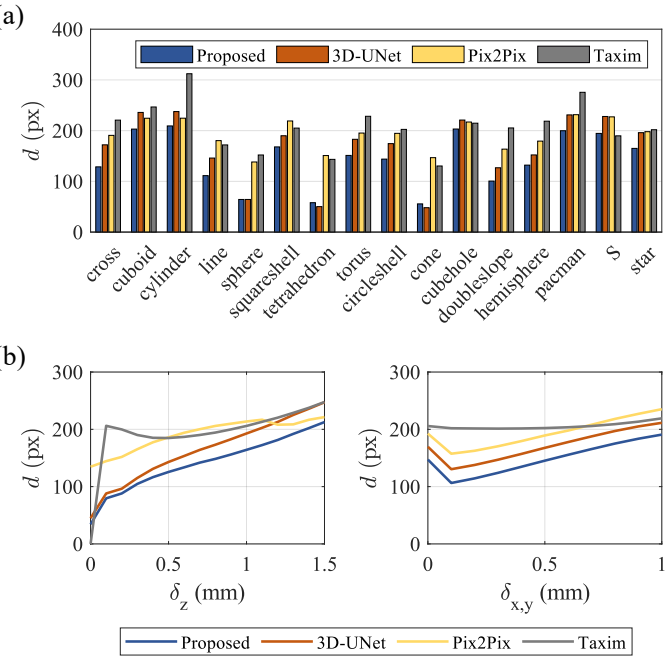

Fig. 8. (a) Sum of marker positional error d of images from each contacted object. (b) Graph of d with respect to normal deformation δ_z and shear deformation $\delta_{x,y}$.

ϵ_{EVM} of marker motion vectors by taking the sum of the magnitude of the error vectors between the real and generated marker motion vectors and expressing it as follows:

$$\epsilon_{\text{EVM}} = \sum_{i=1}^{n} \|\mathbf{v}_{i_{gen}} - \mathbf{v}_{i_{real}}\|, \quad (7)$$

where $\mathbf{v}_i$ is the motion vector of the i^{th} marker. We defined degenerate images as those that could not fully reconstruct markers that are detectable through RGB thresholding and blob detection.

Table I summarizes the results of marker generation performance evaluations for different models. Our method generated markers with a mean positional error sum of 147 px (approximately 2.3 px error per marker). This is equivalent to 0.144 mm error per marker. Our model also shows the lowest ϵ_{EVM} value, meaning that it generates more accurate marker motion vectors. The proposed model failed to generate tactile images with fully detectable markers at an average rate of 0.52 %, but this rate is significantly lower than that of other GANs.

Fig. 8 (a) compares the sum of marker position errors for tactile images with different contacted objects. We used the eight objects on the left to create both the train and test data. The eight on the right are used to create only the test data. No clear difference can be found in marker errors between the two groups. Hence, inter- and intra-object generalization performances are similar for all models. Generally, images from contacting objects with larger areas (e.g., *cylinder* and *pacman*) show a greater marker position error, as more markers are displaced and are, thus, more likely to accumulate more positional errors. Markers generated by the proposed model have the smallest error at all deformation levels (Fig. 8 (b)).

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.

TABLE II
MEAN AND STANDARD DEVIATION (IN BRACKETS) OF IMAGE SIMILARITY METRICS BETWEEN REAL AND GENERATED TACTILE IMAGES FOR MARKER-EXCLUDED REGIONS

	MAE	MSE	SSIM	PSNR
Phong [19]	14.8	354	0.913	22.7
Pix2Pix	12.6 (7.9)	355 (393)	0.908 (0.010)	25.1 (4.6)
3D-UNet	5.01 (2.10)	62.0 (46.9)	0.916 (0.029)	32.2 (3.0)
Proposed	**3.08 (0.13)**	**22.9 (2.8)**	**0.943 (0.001)**	**35.2 (0.4)**

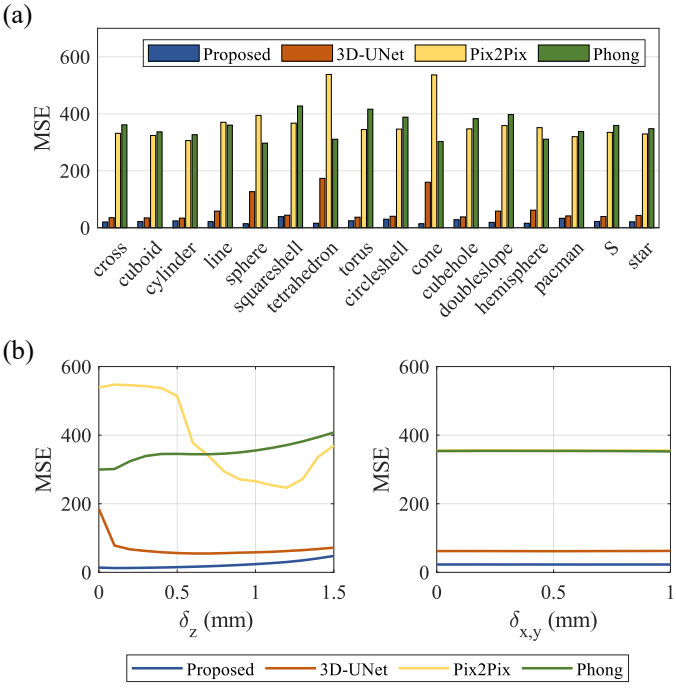

Fig. 9. (a) MSE of non-marker regions of images from each contacted object. (b) Graph of MSE with respect to normal deformation δ_z and shear deformation $\delta_{x,y}$.

Both the proposed and 3D-UNet models demonstrate greater errors when encountering larger deformations. However, the notable distinction between the two models lies in the rate at which the errors escalate as the normal deformations increase. The marker positional error of 3D-UNet approaches the error levels similar to Taxim at large deformations.

B. Image Similarity

We use four image similarity metrics to evaluate and compare the regions of the generated images corresponding to the reflective layer of the elastomer: Pixel-wise mean absolute error (MAE), MSE, structural similarity index measure (SSIM), and peak signal-to-noise ratio (PSNR). While lower MAE and MSE values correspond to better image similarity, the reverse is true for SSIM and PSNR metrics. As the non-marker regions are investigated, we mask out the markers and assess the metrics only for the remaining regions.

Table II shows the image similarity metrics of images generated by different models. Images generated from the proposed model show the best performance on all metrics. Fig. 9 compares the MSE metric of images generated by the different methods in greater depth. Images generated by the proposed model have lower MSE values than those from the other methods across all objects (Fig. 9 (a)) and with respect to the amount of deformation (Fig. 9 (b)). The images from the proposed and 3D-UNet model follow a similar increasing trend with increasing δ_z (Fig. 9 (b)). However, results from the 3D-UNet show a conspicuously high MSE value compared to those from the proposed model at $\delta_z = 0$.

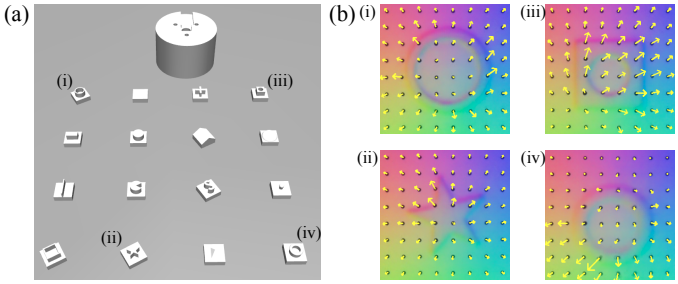

Fig. 10. (a) Object sliding setup. Unfixed objects are intentionally rotated for clearer visualization. (b) Generated tactile images with marker motion vectors after the sliding motion. (i) and (ii) are images from sliding unfixed objects, while (iii) and (iv) are those from sliding against fixed objects. The marker motion vectors are scaled up by 5 for clearer visualization.

C. Application: Object sliding

To demonstrate how the marker-embedded tactile generator can be used in a manipulation task, we set up an environment based on the simulator setup shown in Fig. 5 (d). The objective of the task is to slide each object using the simulated sensor to determine which objects are fixed and unfixed to the table (Fig. 10 (a)). We used the fact that shear displacements would occur on the virtual elastomer when sliding against a fixed object, while when sliding on unfixed objects, objects would move with the sensor, causing negligible displacements.

For each object, the sensor pushed itself down onto the object by 1 mm and moved laterally by 1 mm. During this process, two tactile images were generated: one after the normal motion and the other after the lateral sliding motion (Fig. 10 (b)). We computed the difference between the sum of marker motion vectors of the two images, ΔM, as follows:

$$\Delta M = \left\| \sum_{i=1}^{64} \mathbf{v}_{i_{normal}} - \sum_{i=1}^{64} \mathbf{v}_{i_{slide}} \right\|, \tag{8}$$

where $\mathbf{v}_{i_{normal}}$ and $\mathbf{v}_{i_{slide}}$ are the marker motion vector of the i^{th} marker of the two tactile images, respectively. ΔM was thresholded to identify which objects exerted shear displacements on the sensor. With the exception of the two particularly small objects, *cone* and *tetrahedron*, we were able to identify the unfixed object correctly in all cases.

VI. DISCUSSION AND CONCLUSION

We proposed a GAN-based method to generate realistic marker-embedded tactile images. Our method directly translates simulated depth images into RGB tactile images without needing separate pipelines for generating the reflective layer and marker images. Our proposed method can integrate previous deformation sequences to discriminate deformations from different directions, upon which the marker motions are varied.

We validated the performance of the trained generator using two types of assessment: errors in generated markers and image similarity. The results show that our method outperforms the illumination model and FEM-based method.

The foremost future work is to use our method for the simulation and sim-to-real transfer of other various contact-rich manipulation tasks, especially those requiring shear force or slip detection, deducible from marker motions (e.g., peg-in-hole and assembly tasks). Another possible future work is to apply our GAN-based method to differing sensor configurations by either training generators for each domain or modifying the GAN such that the generator is capable of multidomain I2I. Additionally, other emerging generative methods (e.g., diffusion models), which have shown outstanding performances in other image domains, should be explored in generating marker-embedded tactile images.

REFERENCES

[1] Q. Li, O. Kroemer, Z. Su, F. F. Veiga, M. Kaboli, and H. J. Ritter, "A review of tactile information: Perception and action through touch," *IEEE Transactions on Robotics*, vol. 36, no. 6, pp. 1619–1634, 2020.

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

[3] A. Billard and D. Kragic, "Trends and challenges in robot manipulation," *Science*, vol. 364, no. 6446, p. eaat8414, 2019.

[4] W. Yuan, S. Dong, and E. H. Adelson, "Gelsight: High-resolution robot tactile sensors for estimating geometry and force," *Sensors*, vol. 17, no. 12, p. 2762, 2017.

[5] I. H. Taylor, S. Dong, and A. Rodriguez, "Gelslim 3.0: High-resolution measurement of shape, force and slip in a compact tactile-sensing finger," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 10 781–10 787.

[6] 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)*, 2020, pp. 618–624.

[7] 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*, vol. 5, no. 3, pp. 3838–3845, 2020.

[8] B. Ward-Cherrier, N. Pestell, L. Cramphorn, B. Winstone, M. E. Giannaccini, J. Rossiter, and N. F. Lepora, "The tactip family: Soft optical tactile sensors with 3d-printed biomimetic morphologies," *Soft Robotics*, vol. 5, no. 2, pp. 216–227, 2018.

[9] A. Alspach, K. Hashimoto, N. Kuppuswamy, and R. Tedrake, "Soft-bubble: A highly compliant dense geometry tactile sensor for robot manipulation," in *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*, 2019, pp. 597–604.

[10] W. Kim, W. D. Kim, J.-J. Kim, C.-H. Kim, and J. Kim, "Uvtac: Switchable uv marker-based tactile sensing finger for effective force estimation and object localization," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6036–6043, 2022.

[11] M. K. Johnson and E. H. Adelson, "Retrographic sensing for the measurement of surface texture and shape," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1070–1077.

[12] S. Dong, W. Yuan, and E. H. Adelson, "Improved gelsight tactile sensor for measuring geometry and slip," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 137–144.

[13] D. Ma, E. Donlon, S. Dong, and A. Rodriguez, "Dense tactile force estimation using gelslim and inverse fem," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 5418–5424.

[14] S. Dong, D. K. Jha, D. Romeres, S. Kim, D. Nikovski, and A. Rodriguez, "Tactile-rl for insertion: Generalization to objects of unknown geometry," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 6437–6443.

[15] C. Wang, S. Wang, B. Romero, F. Veiga, and E. Adelson, "Swingbot: Learning physical features from in-hand tactile exploration for dynamic swing-up manipulation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 5633–5640.

[16] J. Lin, R. Calandra, and S. Levine, "Learning to identify object instances by touch: Tactile recognition via multimodal matching," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 3644–3650.

[17] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 23–30.

[18] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 3803–3810.

[19] D. F. Gomes, P. Paoletti, and S. Luo, "Generation of gelsight tactile images for sim2real learning," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 4177–4184, 2021.

[20] 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*, vol. 7, no. 2, pp. 3930–3937, 2022.

[21] 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)*, 2021, pp. 1–7.

[22] Y. She, S. Wang, S. Dong, N. Sunil, A. Rodriguez, and E. Adelson, "Cable manipulation with a tactile-reactive gripper," *The International Journal of Robotics Research*, vol. 40, no. 12-14, pp. 1385–1401, 2021.

[23] Z. Si and W. Yuan, "Taxim: An example-based simulation model for gelsight tactile sensors," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2361–2368, 2022.

[24] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, p. 139–144, oct 2020.

[25] S. Cotin, H. Delingette, and N. Ayache, "Real-time elastic deformations of soft tissues for surgery simulation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 5, no. 1, pp. 62–73, 1999.

[26] Y. Pang, J. Lin, T. Qin, and Z. Chen, "Image-to-image translation: Methods and applications," *IEEE Transactions on Multimedia*, vol. 24, pp. 3859–3881, 2022.

[27] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige, S. Levine, and V. Vanhoucke, "Using simulation and domain adaptation to improve efficiency of deep robotic grasping," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 4243–4250.

[28] J. Zhang, L. Tai, P. Yun, Y. Xiong, M. Liu, J. Boedecker, and W. Burgard, "Vr-goggles for robots: Real-to-sim domain adaptation for visual control," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1148–1155, 2019.

[29] S. Lin, F. Qin, Y. Li, R. A. Bly, K. S. Moe, and B. Hannaford, "Lc-gan: Image-to-image translation based on generative adversarial network for endoscopic images," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 2914–2920.

[30] Y. Bang, Y. Lee, and B. Kang, "Image-to-image translation-based data augmentation for robust ev charging inlet detection," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3726–3733, 2022.

[31] 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)*, 2022, pp. 8305–8311.

[32] 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*, vol. 7, no. 3, pp. 6187–6194, 2022.

[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)*, 2017, pp. 2242–2251.

[34] 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)*, 2017, pp. 5967–5976.

[35] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 5026–5033.

[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2024, Yokohama, Japan. Cite as RA-L paper.