Simulation, Learning, and Application of Vision-Based Tactile Sensing at Large Scale

Quan Khanh Luu, *Student Member, IEEE*, Nhan Huu Nguyen, *Member, IEEE*, and Van Anh Ho, *Senior Member, IEEE*

***Abstract*—Large-scale robotic skin with tactile sensing ability is emerging with the potential for use in close-contact human–robot systems. Although recent developments in vision-based tactile sensing and related learning methods are promising, they have been mostly designed for small-scale use, such as by fingers and hands, in manipulation tasks. Moreover, learning perception for such tactile devices demands a huge tactile dataset, which complicates the data collection process. To address this, this study introduces a multiphysics simulation pipeline, called *SimTacLS*, which considers not only the mechanical properties of external physical contact but also the realistic rendering of tactile images in a simulation environment. The system utilizes the obtained simulation dataset, including virtual images and skin deformation, to train a tactile deep neural network to extract high-level tactile information. Moreover, we adopt a generative network to minimize sim2real inaccuracy, preserving the simulation-based tactile sensing performance. Last but not least, we showcase this sim2real sensing method for our large-scale tactile sensor (*TacLink*) by demonstrating its use in two trial cases, namely, whole-arm nonprehensile manipulation and intuitive motion guidance, using a custom-built tactile robot arm integrated with TacLink. This article opens new possibilities in the learning of transferable tactile-driven robotics tasks from virtual worlds to actual scenarios without compromising accuracy.**

***Index Terms*—Deformable model, machine learning, soft robotics, tactile sensors.**

I. Introduction

THE sense of touch not only provides a diverse range of information from interactions involving physical contacts, such as interactive force, texture, and temperature, but also is considered to be a means of communication in human–human or human–machine interactions. Skin, the largest organ of the human body covering whole limb or torso, possesses a tactile sensing system that has been inspiring robotics community toward the creation of fully autonomous social and task-based machines with the sense of touch [1], [2]. For years, research on this topic, especially on the large-scale mimicking human skin, based on various transducing principles has been intensively investigated [3], [4], [5]. Nonetheless, designing tactile sensors faced complexity in system integration and data processing, since increasing the scale requires a great deal of embedded sensing elements. Recently, vision-based tactile (ViTac) sensors have emerged as an effective method for the implementation of tactile sensing with a simple design [6], [7], [8]. In detail, the deformation of soft artificial skins upon physical contact with an object is detected through the optical tracking of visual features, such as markers or reflective membranes, which is then translated into tactile information, including contact location, force, vibration, object texture, and so on. The ViTac sensors have been found useful in small-scale manipulation tasks using robotics hands/fingers [9], [10]; however, their potential uses in large-scale whole-arm applications have not been comprehensively investigated.

We previously demonstrated marker-based vision-based tactile sensing (TacLink) with the potential to deliver rich contact information from tactile images based on image processing techniques [11]. In addition, we leveraged the use of a supervised learning method with a high sampling rate for the same setup [12], [13]. The former method, through thorough model analysis and calibration, can yield high sensing performance; its complication in modeling and processing is not widely preferred. On the other hand, data-driven methods like the latter method need a huge amount of data to categorize visual representations, which requires a burdensome experimental data acquisition process [12]. This problem would be magnified in applications with large-scale skin and more complex contact scenarios. As a result, there is an emerging necessity for a pipeline that allows simulation-based learning and accommodates the physics of interactive contact in ViTac sensing systems. While visual effects have been reflected successfully in several simulators, such as Gazebo or Unity, interactions between the sensor skin and its external environment are often modeled as rigid contacts [14], [15].

In this article, we propose a novel simulation pipeline toward a framework for a large-scale marker-cum-vision-based tactile sensor [see Fig. 1(a)] that employs the physics engine *SOFA*[1]

Manuscript received 28 August 2022; revised 4 December 2022; accepted 13 February 2023. Date of publication 28 February 2023; date of current version 7 June 2023. This work was supported by Japan Science and Technology Agency's Precursory Research for Embryonic Science and Technology program under Grant JPMJPR2038. This article was recommended for publication by Associate Editor Y. Liu and Editor M. Yim upon evaluation of the reviewers' comments. *(Corresponding author: Van Anh Ho.)*

Quan Khanh Luu and Nhan Huu Nguyen are with the Soft Haptics Labs, School of Materials Science, Japan Advanced Institute of Science and Technology, Nomi 923-1292, Japan (e-mail: quan-luu@jaist.ac.jp; nhnhan@jaist.ac.jp).

Van Anh Ho is with the Soft Haptics Labs, School of Materials Science, Japan Advanced Institute of Science and Technology, Nomi 923-1292, Japan, and also with the Japan Science and Technology Agency, Kawaguchi 332-0012, Japan (e-mail: van-ho@jaist.ac.jp).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TRO.2023.3245983.

Digital Object Identifier 10.1109/TRO.2023.3245983

[1]Simulation Open Framework Architecture: www.sofa-framework.org

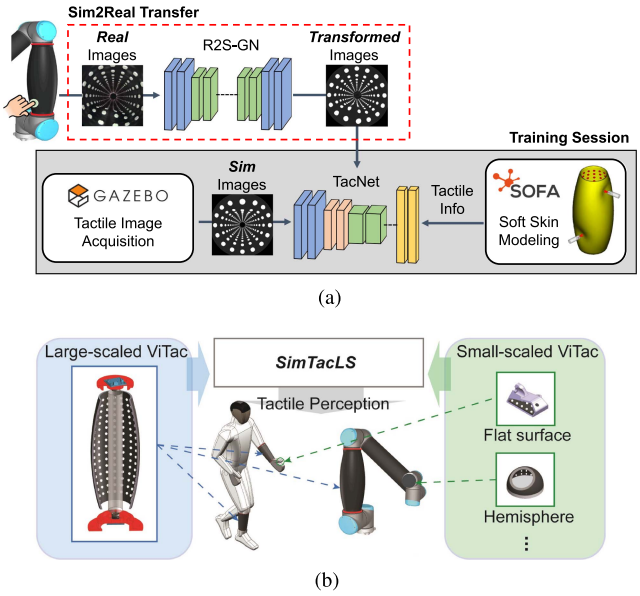


Fig. 1. *SimTacLS* overview. (a) *SimTacLS:* Simulation pipeline for large-scale ViTac sensors. Simulation pipeline, comprised of physics engines SOFA and Gazebo, was constructed to collect a labeled simulation dataset to train the TacNet model, including the information of tactile skin deformation (output) and virtual images (input), and a scheme of sim2real transfer learning was done through a generative network (R2S-GN) of real images into simulation ones. (b) Scalability and extendibility of *SimTacLS* platform. Expected applications of *SimTacLS* to vision-based tactile sensors of diverse shapes and sizes.

to describe complex physical interactions of deformable bodies based on the finite-element method (FEM). The geometry of the skin and the markers (.STL format) were separately fed to *Gazebo* environment, in which a built-in sensor plug-in was exploited to reproduce imaging patterns (i.e., markers) as *visual* versions of tactile images captured by cameras. The virtual images and ground-truth tactile feedback (such as forces and displacements distributed at each element node) recorded from simulation are then utilized as input and output (label) data to train a deep neural network, named *TacNet*. Moreover, in order to apply the effectiveness of the TacNet model to real-world tactile images, we proposed a *real-to-simulation generative* network (R2S-GN) that uses a generative adversarial network (GAN) to automatically learn how to transform real observations of tactile images into simulated ones for evaluation by the TacNet model [see Fig. 1(a)]. Such a platform is envisaged as a standardized easy-to-apply procedure for a wide range of robotic devices at variant scales to acquire tactile perception [see Fig. 1(b)].

II. Related Work

A. Simulation Frameworks for Vision-Based Tactile Sensors

In the literature to date, the two dominant groups of ViTac sensors rely on 1) the intensity of reflective light and 2) relative positions of visual markers to deduce tactile information, represented by GelSight [16] and TacTip [17] sensors, respectively. Recently, advanced simulation tools have been emerging, which reduce the burden in acquisition of real data for learning framework and sim2real transfer approaches to facilitate simulated tactile perception. For instance, to capture depth maps of in-contact objects in GelSight images, Gomes et al. [18] used the robotics simulator Gazebo, which provides a simulated depth camera. Depth-based simulated images were then retouched with a Gaussian filter and Phong's rendering model to approximately recreate the complex inter-reflection of light sources upon the deformation of a membrane. Another sim2real technique applied for GelSight-like sensors was reported in [19], where tactile images were accurately generated with the help of physics-based rendering. Wang et al. [14] established TACTO—a promising open-source simulation layout harmonizing physics engine PyBullet and rendering engine Pyrender, where rigid contacts were employed for depth-map-based RGB tactile image rendering at hundreds of frames per second. The feasibility of this simulator was validated with two popular versions of finger-sized ViTac sensors: hemisphere-shaped OmniTact [20] and DIGIT [21], which provided tactile images equivalent to simulated interaction from dynamic simulation. However, a limitation was that cases involving a large deformation of skin upon ViTac contact with its environment were not rendered properly. This issue is crucial in complex contact scenarios and reduces the accuracy of the sim2real transfer process.

Regarding marker-based ViTac sensors that are more comparable to our work, a bottleneck occurs in emulating the deformation of the soft skin by the movement of multiple markers upon external stimulation, which requires a comprehensive modeling of contact mechanics and materials. Ding et al. [22] built and emulated the elastic behavior of TacTip skin using Unity physics engine to estimate pin locations. However, the elastic properties of the skin were linearly approximated using a custom linear elastic model. Church et al. [15] proposed Tactile Gym that produces the virtual images of physical contacts through depth imprints using a rigid contact model. Tactile Gym was validated against finger-sized TacTip sensors of either hemispherical or rectangular shapes, in which the gap between real and virtual images was mitigated using a trained generative framework. Alternatively, commercial FEM-based simulators (e.g., Abaqus [23], [24]) offer a systematic way to accomplish this challenge by dividing the soft body into many subelements, which are then dynamically analyzed with hyperelastic material models. However, extreme computational costs and inflexibility of the commercial FEM simulators restrict the effective application of these methods in real-time scenarios.

B. Simulation-to-Real Transfer by Adversarial Learning

Deep-neural-network-based vision systems learned from virtual/synthetic images typically perform poorly when evaluated on real visual inputs [25]. Differences between simulation and real images are inevitable, which might involve unrealistic texture, color, and lighting conditions in the simulated results. Several previous studies introduced randomization for such visual aspects in simulation environments to narrow the sim2real gap, which has been successfully applied to vision-based robotic applications [26] and tactile sensing devices [18], [22]. Although these techniques enable a trained model to be more robust when transferred to a real dataset domain, they often require

the manual input of visual features to randomize and need to be refined for specific tasks in unique environments. To address this, recently, a concept of pixel-level domain-adaptation-based sim2real transfer has been applied to the simulation framework for a small-scale marker-based tactile sensor [15]. Here, the researchers employed an auxiliary GAN [27] to translate real marker-based tactile images into depth-based simulation ones, on which deep learning agents were trained to perform specific tasks. Despite having shown successes in transferring a handful of high-level tasks with tactile sensory feedback, this method employed real-to-simulation (real2sim) translation for virtual tactile imprints (depth-based images), while real ones featured white markers that encoded high deformation of artificial skin. This may result in an inaccurate reproduction of the entire deformed state of the artificial skin at one local imprint. In addition, the accuracy of this real2sim translator on the entire real image domain has not been thoroughly evaluated.

In this study, we propose a promising simulation framework (*SimTacLS*) for ViTac sensors that is able to address the aforementioned demerits. The main contributions are pictured in Fig. 1 and highlighted as follows:

1) proposal of a simulation framework built from two kernels: SOFA and Gazebo, with a justified pipeline that allows a detailed investigation of marker-based ViTac sensors of diverse shapes and sizes;
2) the deployment of a supervised-learning-based multioutput regression model (*TacNet*), which takes virtual tactile images under various contact scenarios provided by the above framework as input to perform *3-D skin-shape reconstruction* with high spatial–temporal resolution;
3) to mitigate TacNet's incompatibility owing to using real tactile images in actual sensor operation, a real-to-simulation conversion approach (*R2S-GN*) is introduced;
4) the demonstration of the application of this sim2real learning approach to a large-scale tactile sensor (*TacLink*) in tactile-based tasks using a custom-built robot arm with TacLink as a forearm link.

III. SimTacLS: A Simulation Framework for a Large-Scale Vision-Based Tactile Sensor

A. Hardware Design

This section describes the detailed mechanics of SimTacLS. We chose our previously developed large-scale ViTac sensor for the robot link (called TacLink [12], [13]), which uses the displacement of visual cue markers to deduce tactile sensory information, to demonstrate the proposed framework. The structure of this system is shown in Fig. 2(a). The details of the geometrical specifications of the skin on a barrel shaped body are: 260 mm high (or long depending on orientation); 3.5 mm thick; 36-mm-diameter cross-sectional area at each end; and 53.5-mm-diameter cross-sectional area at the widest point (the center). The body contains 256 white markers of diameter $\phi_{\text{marker}} = 5$ mm distributed on the inner wall of the soft skin. The distances between the outermost row of markers to the adjacent row and the small end of the skin are 15 and 17 mm, respectively.

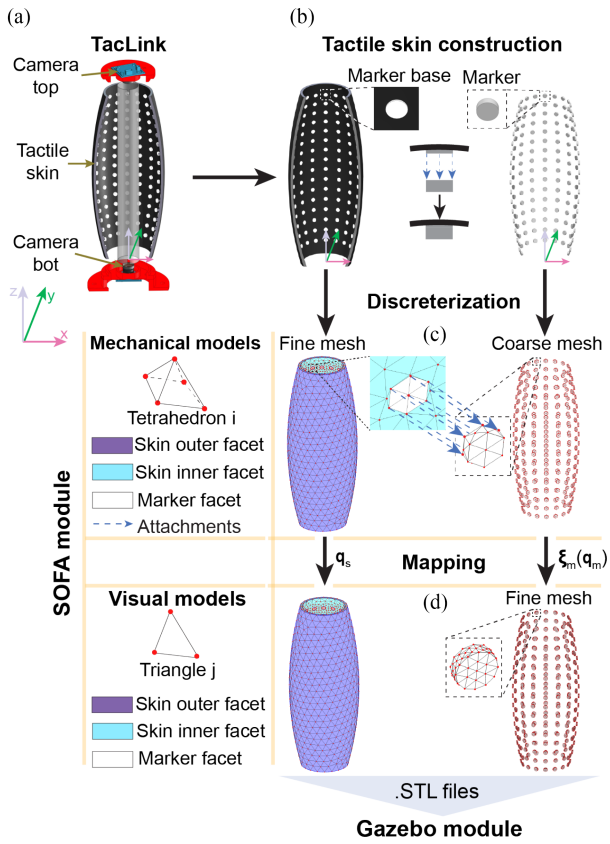


Fig. 2. (a) Hardware architecture of barrel-shaped TacLink (see [12] and [13] for more details on the geometrical design, constituent components, and fabrication process). (b) Cylinder markers attached on the tactile skin will be decomposed into two parts: marker bases and bodies. (c) Each tactile skin element will be imported to SOFA as a topological map of (c) tetrahedron elements for mechanical models and (d) triangular cells for visual models. Notice that while the high quality of the skin mesh still remains in this mode, the meshes for markers in visual model are refined significantly.

We have two reasons for using TacLink as the showcase in this article: 1) the barrel-shaped sensor can be scaled to other parts of a robot body, such as arms, legs, and chest; and 2) this shape allows a rare setup of cameras (opposing), which is considered challenging since large deformation of the skin may prevent cameras from capturing images of all the markers (occlusion). Therefore, the solution of this setup can be applied to many other designs of marker-based ViTac sensors [see Fig. 1(b)].

B. SOFA Module: Skin Reconstruction and Modeling Strategy

Within the SOFA environment, a mechanical model of TacLink skin comprises two separate models: bare skin and markers. These are then consolidated to each other in the simulation environment [see Fig. 2(b) and (c)]. Since SOFA allows the simulation of multiple meshes (with different objectives), the following discretization strategy is obeyed: 1) meshes for studying mechanical behaviors and visualization (visual models) of the bare skin must be sufficiently fine (skin size element is 12 mm); meanwhile, the contrary (marker size element is 1.5 mm) is applied for markers to reduce computational costs. Then, the positions of each of the degrees of freedom (DoFs) in

visual models are inherited from the mechanical models through a mapping function $\boldsymbol{\xi}_m$ before exporting (.STL files) to the Gazebo module.

1) Corotational FEM Approach: The softness of TacLink skin derives from the inherently nonlinear property of soft materials that always pose a critical challenge to mechanical modeling. One can tackle this issue with hyperelastic material models available in off-the-shelf simulation platforms [24], [28]. However, it requires tremendous effort to accurately identify all the necessary parameters through either experimental or numerical processes. Moreover, since we aimed to implement the proposed framework in real-time robotic applications, a method with high efficiency in computation was essential. In this work, connectivity among the vertices of nonoverlapping tetrahedron elements obey a linear constitutive relationship (Hooke's laws) as ascribed by two parameters: Young's modulus E and Poisson's ratio ν. E was experimentally identified as 0.1 N/mm^2 and ν was set at 0.49 [29]. To prevent any unrealistic simulation results due to this linear assumption, especially with large deformations (not only large displacement but also rigid rotation), a corotational FEM formulation was leveraged (see [30] for details). This allows a realistic simulation that captures the geometric nonlinearity of a hyperelastic material (i.e., small stresses produce large strains) in a cost-effective manner.

2) Dynamic Analysis: The generic dynamic equation for a deformable volume is shown as follows:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} = \mathbf{F}^{\text{ext}}(t) - \mathbf{F}^{\text{int}}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{J}^T\boldsymbol{\lambda} \tag{1}$$

where $\mathbf{q} \in \mathbb{R}^n$ is the 3-D position of element nodes (corresponding to N DoFs), $\mathbf{M}(\mathbf{q})$ is the mass matrix, $\mathbf{F}^{\text{ext}}(t)$ denotes the external forces (e.g., gravity) at each time step t, and $\mathbf{F}^{\text{int}}(\mathbf{q}, \dot{\mathbf{q}})$ represents internal forces upon the system state. Equation (1) is integrated over a specific time interval $[t_1, t_2]$, thus $dt = t_2 - t_1$, using the backward Euler integration scheme [31]

$$\mathbf{M}(\dot{\mathbf{q}}_2 - \dot{\mathbf{q}}_1) = dt\left(\mathbf{F}^{\text{ext}}(t_2) - \mathbf{F}^{\text{int}}(\mathbf{q}_2, \dot{\mathbf{q}}_2) + \mathbf{J}^T\boldsymbol{\lambda}\right). \tag{2}$$

Substituting the linearization of the internal forces $\mathbf{F}^{\text{int}}(\mathbf{q}_2, \dot{\mathbf{q}}_2)$ by using Taylor series expansion with the first-order approximation (see [32] for more details) and two relations $\dot{\mathbf{q}} = \mathbf{q}_2 - \mathbf{q}_1 = dt\dot{\mathbf{q}}_2$ and $\ddot{\mathbf{q}} = \dot{\mathbf{q}}_2 - \dot{\mathbf{q}}_1$ into (2) yields

$$\underbrace{\left(\mathbf{M} + dt^2\mathbf{K} + dt\mathbf{C}\right)}_{\mathbf{A}}\underbrace{\ddot{\mathbf{q}}}_{\mathbf{x}} = \underbrace{-dt^2\mathbf{K}\dot{\mathbf{q}}_1 + dt\left(\mathbf{F}_2^{\text{ext}} - \mathbf{F}_1^{\text{int}}\right)}_{\mathbf{b}} + dt\mathbf{J}^T\boldsymbol{\lambda} \tag{3}$$

where $\mathbf{F}_2^{\text{ext}}$ is the external force at the next time step; $\mathbf{K} = \frac{\partial \mathbf{F}^{\text{int}}}{\partial \mathbf{q}}$ and $\mathbf{C} = \frac{\partial \mathbf{F}^{int}}{\partial \dot{\mathbf{q}}}$ are stiffness and damping matrices, respectively. The only unknown factor is $\mathbf{J}^T\boldsymbol{\lambda}$, which represents the contribution of tactile interaction under the form of constraints. The Jacobian matrix $\mathbf{J}(\mathbf{q}) = \frac{\partial \boldsymbol{\xi}}{\partial \mathbf{q}}$ gathers the normal and tangential constraint (i.e., contact forces) directions of $\boldsymbol{\lambda}$—equivalent to the magnitude of contact forces projected to the mapped DoF. Note that the above contact responses obey a combination of Signorini's frictionless contact law [33] and Coulomb's frictional law [34]. This is mathematically expressed by the following complementarity condition:

$$\begin{cases} \text{In contact: } \boldsymbol{\Delta}_n = 0 \Rightarrow \boldsymbol{\lambda}_n > 0 \\ \text{No contact: } \boldsymbol{\Delta}_n > 0 \Rightarrow \boldsymbol{\lambda}_n = 0 \end{cases} \tag{4}$$

where $\boldsymbol{\Delta}_n$ and $\boldsymbol{\lambda}_n$ represent the distance between two contact opponents and the contact force measured along the normal direction n, respectively. Once a contact is well-detected, the contact response $\mathbf{J}^T\boldsymbol{\lambda}$ is computed to determine the above condition. A more in-depth explanation of this particular procedure can be reviewed in [32].

To solve linear equation (3), there are several approaches offered by the SOFA framework. We leveraged the sparse $\mathbf{LDL^T}$ factorization method [33] to decompose matrix $\mathbf{A}$, where $\mathbf{D}$ is a diagonal matrix and $\mathbf{L}$ is a sparse lower triangular part of matrix $\mathbf{A}$. Although this approach is quite costly, the reliability of the simulated mechanical behavior of the soft body (i.e., tactile skin) is assured.

C. Virtual Tactile Image Acquisition

The process for generation and acquisition of virtual (simulated) tactile images is performed using the combination of the *Gazebo* simulator and the Robot Operating System (ROS). The TacLink sensor is modeled as a robotic link using Unified Robot Description Format (URDF),[2] in which the geometric relations between sensor parts, such as housings and cameras, are defined precisely as in the design of a real device. In the URDF description, the *Gazebo* sensor plug-in providing the camera type of `Wide Angle Camera Sensor` is installed to enable virtual cameras to render images of the artificial skin (tactile images). To capture realistic images of skin deformation, the topological meshes of the sensor's soft skin and marker (.STL format) generated from SOFA simulation are updated at each time step and encoded in the SDF format[3] to communicate with the Gazebo environment .

D. TacNet-Based Skin Shape Reconstruction

TacNet was developed to reconstruct, from a pair of tactile images, the geometrical shape of soft skin deformed by external forces in order to deduce high-level tactile perception. This vision-based reconstruction problem can be formulated as a multioutput regression task: given a pair of marker-featured tactile images $\mathcal{I} = \{I_1, I_2\}$, where I_1 and I_2 are RGB images of size $640 \times 480 \times 3$ pixels, the network estimates the displacement vectors ($\mathbf{D}_{\text{estimated}}$) of nodes $\mathcal{N}$ ($|\mathcal{N}| = 707$ nodes) of a surface mesh representing the soft sensor skin (see Section III-B)

$$\mathbf{D}_{\text{estimated},i} := \mathbf{X}_i - \mathbf{X}_{0,i} \quad \forall i \in \mathcal{M} \tag{5}$$

where $\mathbf{X}_i \in \mathbb{R}^3$ is the 3-D position vector of one active/free node $i \in \mathcal{M}$, where $\mathcal{M}$ is a set of free nodes ($|\mathcal{M}| = n = 585$ nodes), and $\mathbf{X}_{0,i} \in \mathbb{R}^3$ are the coordinates of the respective node under the original or nondeformed state of the artificial skin. Thus, from the estimated displacement vectors $\mathbf{D}_{\text{estimated}}$ and original nodal positions $\mathbf{X}_0$, the skin shape can be reconstructed as $\mathbf{X} =$

[2]URDF is an XML format used by the ROS to describe a robotics system.
[3]Simulation Description Format: http://sdformat.org/

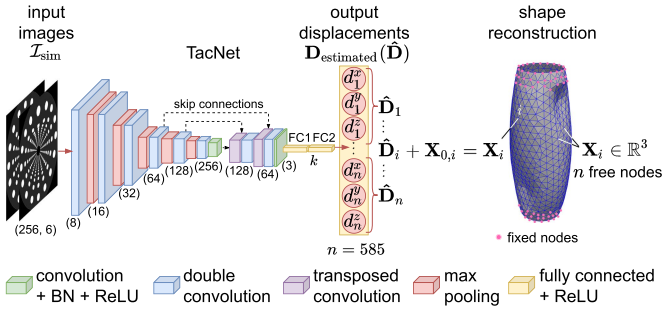


Fig. 3. TacNet concept and architecture. It maps a pair of virtual tactile images $\mathcal{I}_{\text{sim}}$ to the displacements of free nodes $\mathbf{D}_{\text{estimated}}$ from which the shape of artificial skin could be reconstructed. The skin is represented by a topological mesh consisting of fixed nodes (denoted by pink dots) and free ones (the other vertices of triangular cells). The backbone of TacNet is constructed from the Unet network with the modification of downsampled input signal $(256, 6)$ and dense output layer (1755) where every three neurons represents the estimated displacement vector of one free node $\mathbf{D}_{\text{estimated},i}$.

$\mathbf{D}_{\text{estimated}} + \mathbf{X}_0$, with the positions of all fixed nodes always unchanged $\mathbf{X}_i = \mathbf{X}_{0,i}$, $\forall i \in \mathcal{B}$ ($\mathcal{B}$ is a set of fixed nodes, such that $\mathcal{N} = \mathcal{M} \cup \mathcal{B}$).

1) TacNet Architecture: The TacNet architecture is adapted from proven Unet convolution networks [35]. Basically, TacNet consists of a contracted convolution path connected with a reverse upconvolution one via skip connections and then followed by two fully connected (FC) layers (see Fig. 3). For the input layer, we concatenate the two tactile images $\mathcal{I}$, downsampled to 256×256, to form a six-channel input visual signal. Moreover, the output signal, activated by the two last FC layers, is defined by a dense single layer with 1755 neurons to represent the estimated displacement vectors $\mathbf{D}_{\text{estimated}}$, which means that we consider every three adjacent neurons as a displacement vector.

2) TacNet Training and Loss Function: TacNet is trained completely on a *simulation dataset* with the input data $\mathcal{I}_{\text{sim}}$ (pair of images obtained from simulated TacLink cameras) and corresponding output labels $\mathbf{D}_{\text{FEM}}$ (ground-truth displacement vectors) generated, respectively, in *Gazebo*/ROS and SOFA environments (see Sections III-C and III-B). We apply the mean squared error (MSE) loss as an objective function to minimize the differences between the ground-truth and estimated displacement vectors $(\mathbf{D}_{\text{FEM}}, \mathbf{D}_{\text{estimated}})$ and to optimize the weights of TacNet T_θ

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathcal{L}_{\text{MSE}}[\mathbf{D}_{\text{FEM}}, \mathrm{T}_\theta(\mathcal{I}_{\text{sim}})] \tag{6}$$

where $\mathbf{D}_{\text{estimated}} = \mathcal{T}_\theta(\mathcal{I}_{\text{sim}})$ and $\mathcal{L}_{\text{MSE}}(\cdot)$ is MSE loss, given by

$$\mathcal{L}_{\text{MSE}}(\mathbf{D}_{\text{FEM}}, \mathbf{D}_{\text{estimated}}) = \frac{1}{3n} \sum_{i \in \mathcal{N}} \sum_{j \in \{x,y,z\}} (d^j_{\text{FEM},i} - d^j_{\text{estimated},i})^2 \tag{7}$$

where $d^j_i \quad \forall j \in \{x, y, z\}$ are the components of displacement vector $\mathbf{D}_i$ at the respective skin node $i \in \mathcal{M}$ along the x, y, and z axes. In fact, the MSE loss in (7) is derived to compute the difference in every vector component (or output neurons) to encourage the learning of both intensity and direction of displacement vectors. For the optimization (6), we use iterative stochastic gradient descent optimizer with the experimentally tuned learning rate of 0.015.

E. Real-to-Simulation Generative Network

The main purpose of the R2S-GN is to transform real tactile images (I_{real}) into ones (*transformed* images I_{tf}) that resemble visual inputs of the simulation dataset (I_{sim}) before they are fed to TacNet, so the performance of TacNet-based 3-D shape reconstruction is maintained in real-world deployment. For this purpose, the R2S-GN is trained in an adversarial manner, where it plays a role as a generator in a traditional GAN, competing against a discriminator in order to achieve its best in the transformation task.

1) Network Architectures: We exploited the adapted version of the U-Net convolutional network and the PatchGAN model, as described in [36], for the architecture of R2S-GN generator (G_ϕ) and discriminator (D_ψ), respectively. G_ϕ takes as input the downsampled real images (I_{real}, $256 \times 256 \times 3$) on a encoder path and outputs the transformed counterparts (I_{tf}) on a reverse decoder path. Meanwhile, the discriminator (D_ψ) receives a $256 \times 256 \times 3$ pixel input image, and the network classifies whether the images inputted is *real* or *fake*. Details of the network parameters for G_ϕ and D_ψ architectures can be found in [36].

2) R2S-GN Loss Function: We propose a hybrid loss function $\mathcal{L}_{\text{R2S-GN}}$ that is used to train the R2S-GN generative network (G_ϕ). This loss function comprises three terms, including conditional GAN (cGAN) adversarial objective, ℓ_1 distance, and structural similarity (SSIM) loss.

a) Image appearance loss: Inspired by Zhao et al. [37], we introduce an appearance loss that combines ℓ_1 distance with the SSIM metric (for image quality quantification) [38]. This loss function, which evaluates on a pixel-by-pixel basis, is defined to match the appearance of the *fake* transformed images I_{tf} with the *real* simulation ones I_{sim}, as well as to ensure SSIM between them. This is vital to generate images with the same geometric perspective as simulation ones, in order to maintain the skills of simulation-trained TacNet. Thus, for a given batch of one training sample, this loss is given as

$$\mathcal{L}_{\text{img}} = \alpha \left\| \mathrm{G}_\phi(I_{\text{real}}) - I_{\text{sim}} \right\|_1 + \beta \frac{1 - \text{SSIM}(\mathrm{G}_\phi(I_{\text{real}}), I_{\text{sim}})}{2} \tag{8}$$

where we apply a 11×11 Gaussian kernel for the computation of the SSIM metric.

b) Adversarial loss: In addition to the appearance loss, we adopt the cGAN objective [36] for an adversarial loss term. For a given real tactile image I_{real}, the adversarial loss for the R2S-GN G_ϕ can be expressed as

$$\mathcal{L}_{\text{adv}} = \log\left(1 - \mathrm{D}_\psi\left(I_{\text{real}}, \mathrm{G}_\phi(I_{\text{real}})\right)\right) \tag{9}$$

where D_ψ, besides observing the transformed version of tactile image $I_{\text{tf}} = \mathrm{G}_\phi(I_{\text{real}})$, is conditional on the input of G_ϕ, particularly I_{real}. This conditional discriminator has been shown to improve the performance of numerous image translation

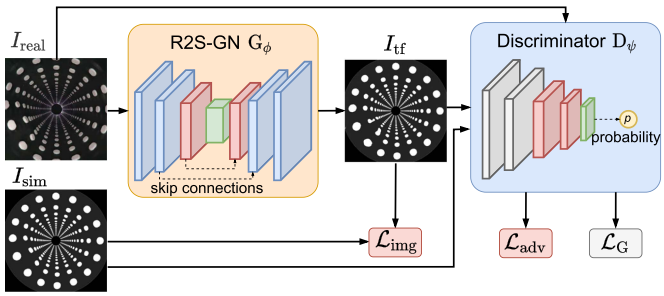


Fig. 4. Training scheme for the R2S-GN model, mostly following the procedure described in [27], however, with the modification for the inclusion of R2S-GN loss (see Section III-E3 for details).

tasks [36]; thus, we applied this concept in our real2sim network. Intuitively, the R2S-GN G_ϕ tries to minimize this objective function by generating the transformed image that can fool the adversarial discriminator D_ψ into predicting it as a *real* simulation image. As a result, the overall loss objective for R2S-GN G_ϕ is the combination of appearance loss as well as the cGAN criteria

$$\mathcal{L}_{\text{R2S-GN}} = \underbrace{\mathcal{L}_{\text{img}}}_{\text{Appearance loss}} + \underbrace{\gamma \cdot \mathcal{L}_{\text{adv}}}_{\text{Adversarial loss}} \tag{10}$$

where we set the hyperparameters $\alpha = 100$, $\beta = 200$, and $\gamma = 1$, which are tuned experimentally.

Finally, for training the adversarial discriminator D_ψ, we use the cGAN objective as described in [36]. For one training sample, the discriminator loss is given as

$$\mathcal{L}_{\text{G}} = \log\left(1 - D_\psi(I_{\text{real}}, I_{\text{sim}})\right) + \log D_\psi\left(I_{\text{real}}, G_\phi(I_{\text{real}})\right). \tag{11}$$

The second term characterizes the adversarial training behavior where the discriminator tries to maximize the R2S-GN's adversarial objective [see (9)]; meanwhile, the R2S-GN attempts to minimize it. The overall loss [see (11)] indicates that the discriminator would do its best at discriminating the transformed images I_{tf} with simulation ones I_{sim}, which, in turn, penalizes the R2S-GN to generate I_{tf} that more closely match the appearance of I_{sim}.

3) R2S-GN Training: We follow the typical procedure of adversarial GAN training [27] for optimizing the weights of the R2S-GN G_ϕ (see Fig. 4). Specifically, for discriminator training, we set its output label to a positive class (*real*) given that the input is a simulation image, and to negative class (*fake*) provided that the input is a transformed one. As for the R2S-GN, in addition to computation of the $\mathcal{L}_{\text{img}}$ loss, the output label of D_ψ is set to the *real* positive class in order to promote the adversarial $\mathcal{L}_{\text{adv}}$ loss [36]. For the learning process, we used the Adam optimizer with linear learning rate scheduling [39], initialized at 0.0002 and set to decay at the 100th iteration out of a total of 200 training steps.

IV. Large-Scale Tactile Perception Methodology

Large-scale ViTac sensors are suitable to offer *multipoint* physical interactions, which embrace new possibilities for tactile interfaces and make them unique compared to their small-sized counterparts (e.g., tactile fingertips). Among information that can be extracted from the multipoint stimuli, the identification of contact locations on an artificial skin body has found practical use in robotics tasks, such as providing feedback signals for a collision handling framework [40]. Therefore, in addition to a simple method for contact event detection (see Section IV-A), we developed an algorithm to identify multiple contact locations on a large-scale skin (see Section IV-B), which are reasoned from the TacNet model.

A. Touch Sensing

The detection of touch/contact events is considered to be fundamental in safety-critical robotic systems [40]. Here, we present a method to extract contact detection signals based on the recognition of skin deformation.

The contact detection problem can be formulated as a binary classification task where given the displacement vectors $\mathbf{D}_{\text{estimated}}$ estimated by TacNet (5), we assign a *contact detection signal*, which is 0 for data without contact and 1 for data with contact detected. Thus, the contact detection signal can be derived as

$$\text{CD} = \begin{cases} 1, & \text{if } \exists i \,:\, \|\mathbf{D}_{\text{estimated},i}\| \geq \epsilon_c \\ 0, & \text{otherwise} \end{cases}. \tag{12}$$

In other words, for each contact detection, we set a contact detection threshold ϵ_c on the estimated displacement magnitude of free skin nodes $\|\mathbf{D}_{\text{estimated},i}\| \;\; \forall i \in \mathcal{M}$, where ϵ_c depends mainly on the accuracy of TacNet estimation, which would influence detection sensitivity and accuracy. The detection threshold is determined such that contact detection performance reaches a good compromise between precision and recall, which are the metrics of a general binary classifier. We use the simulation dataset to determine the detection threshold, which is expected to transfer well to the distribution of real data. The results of this contact classifier are discussed in more detail in Section V-D.

B. Multipoint Contact Localization

This section presents an algorithm that can identify the contact positions at *multiple points* on the sensing link. This detection method assumes that any contacts occurring between the sensor skin and external objects are point contacts, which is considered to be a reasonable assumption in practical applications [40]. In general, the algorithm applies the concept of graph-theory-based connected-component labeling [41] to extract contact regions, which we named *contact region labeling* (CRL), from which the contact positions are identified. Here, we modeled the mesh of artificial skin as an undirected *graph*, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, whose vertices represent the mesh nodes ($|\mathcal{V}| = |\mathcal{N}| = N$) and contain information on the displacement vectors estimated by TacNet ($\mathbf{D}_{\text{estimated}} \in \mathbb{R}^{3\,N}$). Besides this, every graph node contains information of a fixed radial vector pointing toward the central axis of the skin to determine the inward deflected nodes. Thus,

Algorithm 1: Multiple-Point Contact Localization.

Input: $\mathcal{G}$: skin graph defined by $(\mathcal{V}, \mathcal{E})$; $\mathbf{X}_0$: initial nodal positions; $\mathbf{D}_{\text{estimated}}$: estimated nodal displacement vectors; $\mathbf{N}$: nodal radial inward vectors
Output: $\mathbf{X}_c$: multiple contact positions $(\mathbf{x}_1^c, \ldots, \mathbf{x}_L^c)$

Initialize: ϵ_c ▷ contact threshold
$\mathbf{s} \leftarrow$ newList
for each node v_i in $\mathcal{V}$ **do**
  if $\|\mathbf{D}_{\text{estimated},i}\| \geq \epsilon_c$ and $d_{\text{sim}}(\mathbf{D}_{\text{estimated},i}, \mathbf{N}_i) > 0$ **then** $s_i \leftarrow 1$ ▷assign nodal contact signals (14)
  else $s_i \leftarrow 0$
  end if
end for
$\mathbf{y} \leftarrow \texttt{CRL}(\mathcal{V}, \mathcal{E}, \mathbf{s})$ ▷ obtain a list of contact region labels
$(\boldsymbol{R}_1, \ldots, \boldsymbol{R}_L) \leftarrow \texttt{sortContactRegions}(\mathbf{y})$ ▷ see (16)
$(i_1^*, \ldots, i_L^*) \leftarrow \texttt{searchContactNodes}(\boldsymbol{R}, \mathbf{D}_{\text{estimated}})$ ▷ (see 17)
$\mathbf{x}_l^c \leftarrow \mathbf{X}_{0,i_l^*}$ for all $l \in \{1, \ldots, L\}$
return list of contact positions $(\mathbf{x}_1^c, \ldots, \mathbf{x}_L^c)$
function CRL($\mathcal{V}$, $\mathcal{E}$, $\mathbf{s}$) ▷ contact region labeling
  $l \leftarrow 1$ ▷ initialize contact region label
  $\mathbf{y} \leftarrow$ newList
  for each node v_i in $\mathcal{V}$ **do**
    if $s_i = 1$ and $y_i = \varnothing$ **then**
      $\mathbf{y} \leftarrow \texttt{DFS}(l, v_i, \mathcal{E}, \mathbf{s}, \mathbf{y})$
      $l \leftarrow l + 1$
    else if $s_i = 0$ **then** $y_i \leftarrow 0$
    end if
  end for
  return list of contact labels $\mathbf{y}$
end function
function DFS(l, v_i, $\mathcal{E}$, $\mathbf{s}$, $\mathbf{y}$) ▷ depth first search
  if $s_i = 0$ or y_i not $\varnothing$ **then return** $\mathbf{y}$
  end if
  $y_i \leftarrow l$
  for each neighbor node v_j of v_i in $\mathcal{E}(v_i)$ **do**
    $\mathbf{y} \leftarrow \texttt{DFS}(l, v_j, \mathcal{E}, \mathbf{s}, \mathbf{y})$
  end for
end function

we define the radial vector at every node as

$$\mathbf{N}_i := \begin{bmatrix} 0 & 0 & x_{0,i}^z \end{bmatrix}^\top - \mathbf{X}_{0,i}^\top \quad \forall i \in \mathcal{N} \tag{13}$$

where x_0^z is the z-component of nodal positions $\mathbf{X}_0$ in the undeformed state.

To run CRL for the extraction of distinct contact regions, we need to determine which nodes of the skin are likely to be experiencing contact. Accordingly, we define N-tuple of binary *nodal contact signals* $\mathbf{s} = (s_1, \ldots, s_N) \in \mathbb{Z}_2^N$, where its element s_i holds a binary value $s_i \in \{0, 1\}$ such that $s_i = 1$ indicates that the corresponding node $i \in \mathcal{N}$ is in contact and definitely lies in one contact region; otherwise, $s_i = 0$ signifies that the given node remains intact. Specifically, the nodal contact signal for each node $i \in \mathcal{N}$ is derived as

$$s_i = \begin{cases} 1, & \text{if } \|\mathbf{D}_{\text{estimated},i}\| \geq \epsilon_d \wedge d_{\text{sim}}(\mathbf{D}_{\text{estimated},i}, \mathbf{N}_i) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

where

$$d_{\text{sim}}(\mathbf{D}_{\text{estimated},i}, \mathbf{N}_i) = \frac{\mathbf{D}_{\text{estimated},i} \cdot \mathbf{N}_i}{\|\mathbf{D}_{\text{estimated},i}\| \, \|\mathbf{N}_i\|}. \tag{15}$$

In other words, a node is considered to lie in a contact region if its nodal displacement exceeds a constant threshold ϵ_d and the direction of its displacement vector has to be pointing toward the skin central axis. Under contacts mostly due to pushing/pressing actions, the latter condition helps to restrict contact regions to those that contain nodes deflecting inwards, as opposed to those regions that bulge out. This is measured by the directional similarity term $d_{\text{sim}} \in [-1, 1]$ (15), which, in fact, is $\cos\varphi_i$ (where φ_i is the angle between two vectors $\mathbf{D}_{\text{estimated},i}$ and $\mathbf{N}_i$).

Given the skin graph $\mathcal{G}$ and nodal contact signals $\mathbf{s}$, we perform the CRL procedure to extract possible *multiple* distinct contact regions (see Algorithm 1: CRL function). This procedure employs depth-first search (DFS) to traverse across vertices $\mathcal{V}$ of graph $\mathcal{G}$ that contain the corresponding nodal information of $\mathbf{s}$. On the search path, it selectively assigns a *contact region label* $l \in \{1, \ldots, L\}$ (L is the number of contact regions) to every node that holds the signal $s_i = 1$ such that a cluster of contacted nodes (or a contact region), separated from others by undeformed nodes ($s_i = 0$), shares the same region label l. As a result, we can obtain a set of labels $\mathbf{y} = (y_1, \ldots, y_N) \in \mathbb{Z}_{L+1}^N$ whose element $y_i \in \{0, 1, \ldots, L\}$ corresponds to the region label of node $i \in \mathcal{N}$, and $y_i = 0$ marks nodes inside the undeformed region. From $\mathbf{y}$, we can extract contact regions that are represented by the node indices. Thus, for a given contact region $\boldsymbol{R}_l$ that has the region label l, we have

$$\boldsymbol{R}_l = \{i \in \mathcal{N} \mid y_i = l\}, \forall l \in \{1, \ldots, L\}. \tag{16}$$

Finally, for every single contact region $\boldsymbol{R}_l$, we search for the node $i_l^* \in \boldsymbol{R}_l$ that maximizes the displacement magnitude and consider it as a location where a contact occurs

$$i_l^* = \arg\max_{i \in \boldsymbol{R}_l} \|\mathbf{D}_{\text{estimated},i}\| \quad \forall l \in \{1, \ldots, L\}. \tag{17}$$

From that, contact positions $\mathbf{X}_c = (\mathbf{x}_1^c, \ldots, \mathbf{x}_L^c)^\top \in \mathbb{R}^{3L}$ can be identified from the extracted contact regions

$$\mathbf{x}_l^c = \mathbf{X}_{0,i_l^*} \quad \forall l \in \{1, \ldots, L\}. \tag{18}$$

In addition, corresponding contact depths $\hat{d}(\mathbf{x}_l^c)$ can be derived as

$$\hat{d}(\mathbf{x}_l^c) = \left\|\mathbf{D}_{\text{estimated},i_l^*}\right\| \quad \forall l \in \{1, \ldots, L\}. \tag{19}$$

The step-by-step algorithm for multipoint contact sensing is described in Algorithm 1, whose complexity mainly depends on the size of the skin graph $\mathcal{O}(|\mathcal{V}| + |\mathcal{E}|)$. In addition, the spatial resolution is defined by the fineness of the constructed skin mesh, which poses a tradeoff between the resolution and

TABLE I
COLLECTED DATASETS: DATASETS USED IN THIS ARTICLE FOR MODEL TRAINING AND EVALUATION

Subject	Tactile data	Simulation	Real
TacNet	Images + Info	single + double	–
R2S-GN	Images	single	single
Evaluation	Images + Info	single + double	single + double*

*This dataset only contains some special scenarios used to evaluate SimTacLS and multi-contact localization accuracy.

computational costs; the greater the resolution, the more the computational time. Moreover, the assumption of point contact can be relaxed if contacts induce concave deformation of the skin surface, whereby the detector yields an approximated contact position at the node that was displaced the deepest; however, detection accuracy would fall as the contact plane extended. Finally, in cases where regions of multiple contacts overlap such as an event when two discrete contact points are sufficiently close, the detector might deduce the different regions to be a single large contact area. This sensing behavior, mostly affected by the distance between two contact points and the selection of threshold ϵ_d, along with localization accuracy is discussed in Section V-E.

V. PERFORMANCE EVALUATION

To evaluate the performance of the proposed SimTacLS framework, we conducted some experiments on tactile perception. We used a desktop PC (AMD Ryzen Threadripper 3970X Processor) with GPU acceleration (RTX 8000, NVIDIA) for model training and inference. Demonstrations are available for review in supplementary video associated with this article.

A. Data Collection

In this article, we collected several datasets from simulation and actual models for training and to assess the feasibility of SimTacLS. The details are listed in Table I. The contact locations were set among free nodes $\mathcal{M}$; thus, the total sampled points for the $single-contact$ dataset was 585. To test whether the method could achieve tactile perception in complex scenarios with poor prior experience, 500 contact groups consisting of two arbitrary points among $\mathcal{M}$ were made to produce the $double-contact$ dataset. An experimental setup with an identical reference coordination system, as illustrated in Figs. 2 and 5, was constructed to collect real tactile images. The experiment included three motorized linear stages (Suruga Seiki Co., Japan), a rotating motor (Dynamixel XH430-W350-R, ROBOTIS, Inc., USA), and a stepping motor controller (DS102, Suruga Seiki Co., Ltd., Japan), fixed on a testbed (see Fig. 5). The X-axis stage (PG750-L05AG-UA) drives a spherical-head indentor (12 mm diameter) designed to push one node at a time to the desired contact depth on the skin. The contact locations on the skin outer surface were achieved by horizontal movement of the indentor and the rotation of the TacLink sensor, facilitated by a Z-axis linear carrier (KZS18300) and Z-axis rotating motor, respectively. Meanwhile, the Y-axis stage was preadjusted in advance to ensure that the nominal axis of the indentor intersects with the Z-axis of the reference coordinate system (i.e., centerline of the TacLink sensor).

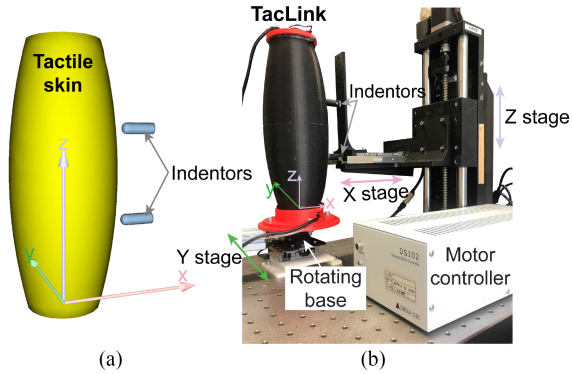


Fig. 5. Setups for data collection in (a) simulation and (b) real world. (a) SOFA environment. (b) Actual setup.

B. Image Transformation With R2S-GN Loss

The performance of the R2S-GN was evaluated by the similarity between pairs of transformed and simulation (baseline) images in terms of spatial image structure. We measured the SSIM index and the complement of per-pixel root mean square error ($\overline{\text{pixRMSE}} = 1 - \text{pixRMSE}$) of the simulation–transformed image pairs. In addition, we compared the performance of the R2S-GN model learned with R2S-GN training objective $\mathcal{L}_{\text{R2S-GN}}$ [see (10)] and the one trained using solely adversarial loss $\mathcal{L}_{\text{adv}}$ and the other with $\mathcal{L}_{\text{adv-L1}} := \mathcal{L}_{\text{adv}} + \mathcal{L}_{\text{L1}}$ (where $\mathcal{L}_{\text{L1}}$ is the ℓ_1-distance loss). The R2S-GN was trained with a total of 18 640 pairs of single-contact actual-simulation images and 4780 image pairs that capture both single (4660 pairs) and double contacts (120 pairs) was devoted for evaluation.

Fig. 6(a) shows the SSIM of tested simulation images with real images transformed by three variants of the R2S-GN (trained, respectively, by the three aforementioned losses) and expresses the variable of their resemblance according to increased contact (touch) depth. In fact, the $\mathcal{L}_{\text{R2S-GN}}$-based R2S-GN generates images that are far more similar to the simulation baselines, providing an average SSIM of 0.96 and an average $\overline{\text{pixRMSE}}$ of 0.95 at 20 mm contact depth, compared, respectively, to 0.91 and 0.90 of the $\mathcal{L}_{\text{adv}}$-based transformation model. Over the observed range of contact depth $d_c \in [1, 20]$, while the former model shows a slight drop in both SSIM and $\overline{\text{pixRMSE}}$ metrics (i.e., around 3.5%), the latter one shows a more significant (7%) drop in SSIM. Moreover, Fig. 6(b) displays representative tested samples of single- and double-contact tactile images with the contact depth of 15 mm. It shows that the unseen tactile images can be generalized and generated well by the R2S-GN even when the model is never trained on the double-contact images, and it once again confirms the effectiveness of the proposed R2S-GN

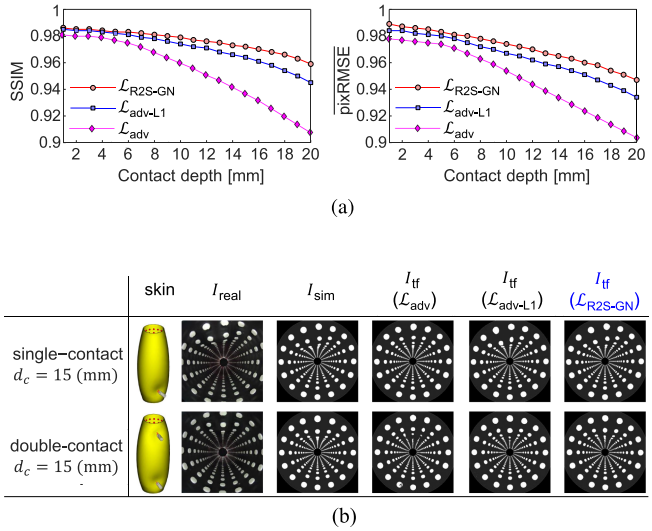
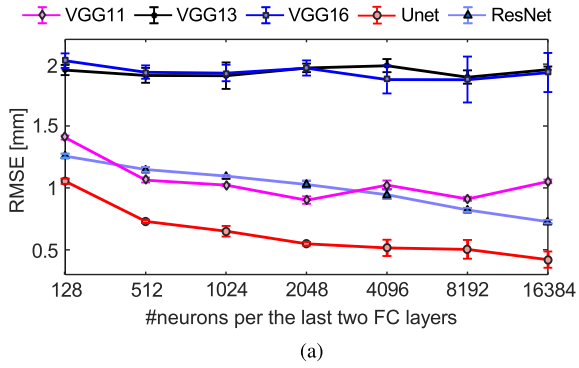


Fig. 6. R2S-GN model evaluation with various training losses. (a) Quantitative simulation–transformed image similarity measured by SSIM and complement of pixRMSE metrics. Spatial similarity between the transformed and *real* simulation images are measured by per-pixel SSIM and pixRMSE metrics (the higher the values, the more the similarity between the compared pairs of images). The graphs present the better performance of the R2S-GN as trained with the proposed $\mathcal{L}_{\text{R2S-GN}}$ loss compared to the other two variants of losses. (b) Representative transformed images (I_{tf}) with variant training objectives comparing to the corresponding ground-truth simulation (I_{sim}) and real images (I_{real}). Visualization of transformed images in the scenarios of single and double contacts ($d_c = 15$ mm).



k	Memory usage (MB)	Inference time (ms)
512	17	4.76 ± 2.49
1024	28	4.72 ± 2.50
2048	54	4.82 ± 2.49
4096	132	4.99 ± 2.49
8192	382	5.45 ± 2.14
16384	1269	6.96 ± 2.66

(b)

Fig. 7. TacNet performance by various network configurations. (a) TacNet performance by configurations. Fivefold cross-validation accuracy of TacNet by varying number of neurons k per the last two FC layers and backbone network architectures. Under the same training conditions, Unet-based TacNet achieves better performance compared to that of the counterparts (smaller RMSE value is better). (b) Specifications of Unet-based TacNet with various number of neurons k. Based on the specifications, it is reasonable to adopt 2048 neurons for the last two FC layers of Unet-based TacNet, which strikes a balance between accuracy, memory usage, and inference time.

loss. In the next subsection, we further evaluate the effectiveness of variant RS2-TN models in addressing the sim2real problem.

C. Sim2Real Transferability of Contact Depth

The performance of the TacNet-based shape reconstruction was verified by evaluating the measurement error of the local contact depth [see (19)], both in simulation and real datasets to prove sim2real effectiveness. For evaluation, Unet-based TacNet with 2048 neurons for each of the last two FC layers was employed, since it was shown, through fivefold cross validation, to outperform other model backbones, such as VGG and ResNet in terms of inference accuracy, speed, and memory usage (see Fig. 7). The used TacNet model was completely trained on the *simulation* dataset including both single- and double-contact images (28 055 pairs of virtual tactile images), in which 20% data of each contact type were withheld as a test fold for validation. For sim2real evaluation, we experimented on a subset of real double-contact images and a full set of real single-contact images corresponding to the simulation test fold (see Table I).

The experimental results showed that measurement errors increased with true contact depth (d_c) in the case of simulation and $\mathcal{L}_{\text{R2S-GN}}$-based translated visual inputs, while pure real ones, without passing through the R2S-GN model, experienced significant errors, yielding estimated values unchanged [see Fig. 8(a) and (b)]. The absolute errors at $d_c = 20$ mm were below 2 and 4 mm, which approximate to full-scale errors 10% and 20% (with FS 20 mm) for simulation and translated inputs, respectively. Fig. 8(c) shows that the $\mathcal{L}_{\text{R2S-GN}}$-based R2S-GN model was superior to the two other variants trained by $\mathcal{L}_{\text{adv}}$ and $\mathcal{L}_{\text{adv-L1}}$, which reduces the full-scale errors of around 25% and 10%, respectively, at $d_c = 20$ mm. In addition, we showcase the visualization of the skin shape reconstruction on two representative scenarios of single- and double-point contact with the depth $d_c = 15$ mm (see Fig. 9). A highly similar sensing pattern between simulation and real (via $\mathcal{L}_{\text{R2S-GN}}$-based R2S-GN) samples was observed in the case of single contact, with an absolute error of around 1.5 mm. In the case of double contact, the mean absolute errors at the two contact patches were 1.31 ± 0.65 mm and 2.92 ± 0.50 mm for the virtual and translated real input, respectively. The occurrence of greater sim2real discrepancies in the double contact was because the R2S-GN had not been trained on double-touch data, which probably results in greater dissimilarity in image structure, especially at large contact depths.

It is worth noting that the sensing performance (specifically, the recognition of contact depth) is dependent on regions of the skin. We conducted an experiment, where the contact was made on various locations (ten locations) along a longitude line of the skin. At each contact location, two contact depth values of 5 and 10 mm were given. The measured data (contact depth value inferred from real images passing through the $\mathcal{L}_{\text{R2S-GN}}$-based R2S-GN) and the ground truths are shown in Fig. 10, revealing that the sensitivity decreases at the equatorial area of the skin. The same issue was actually found in the previous research [11], in which the detection error of around 7% of full scale with

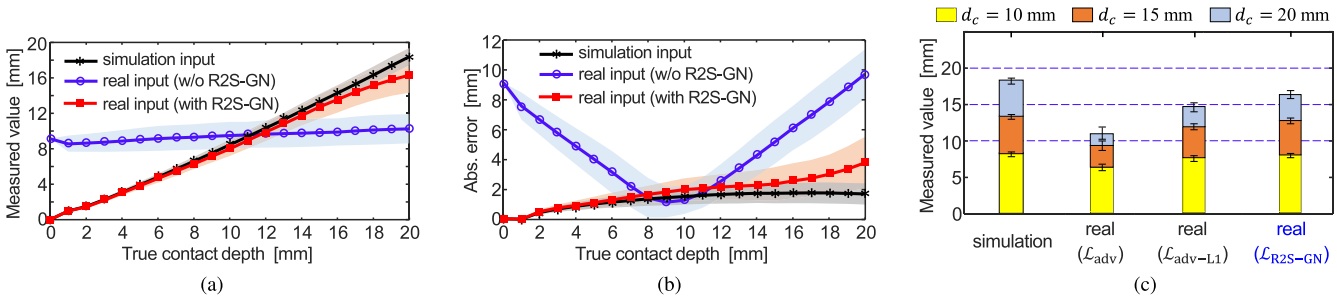


Fig. 8. Evaluation of contact depth accuracy and its sim2real transferability, using the proposed sim2real method. (a) Measured contact depth versus true value. ($\mathcal{L}_{R2S\text{-}GN}$-based R2S-GN was used). (b) Absolute measurement error with increased contact depth. ($\mathcal{L}_{R2S\text{-}GN}$-based R2S-GN was used). (c) Estimated contact depths with various input images compared against true values of 10, 15, and 20 mm.

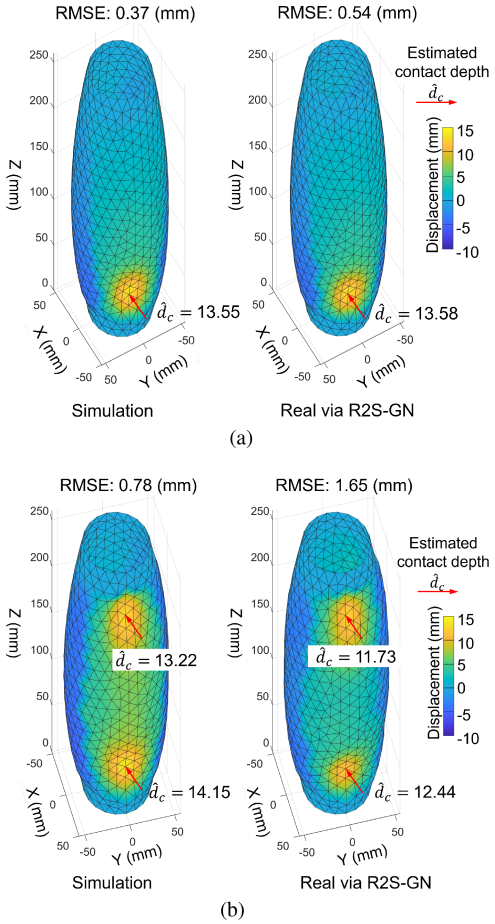


Fig. 9. Visualization of TacNet-based 3-D skin shape reconstruction in the scenarios of single and double contacts with true contact depth at 15 mm. (a) Skin reconstruction with single contact ($d_c = 15$ mm). (b) Skin reconstruction with double contact ($d_c = 15$ mm).

Height of test nodes along TacLink's vertical axis

Node	1	2	3	4	5	6	7	8	9	10
h (mm)	239	219	189	168	138	118	96	65	48	26

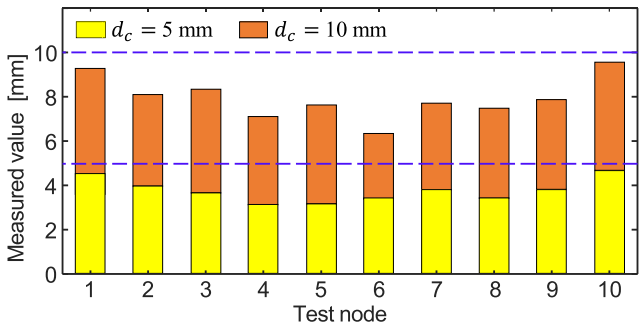


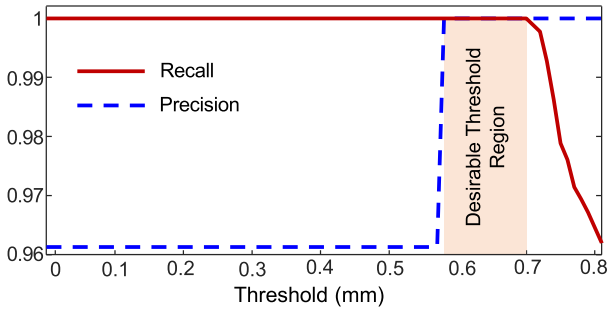
Fig. 10. Evaluation of contact depth estimation at different contact regions on the tactile skin (estimated on real images via $\mathcal{L}_{R2S\text{-}GN}$-based R2S-GN).

the careful calibration of cameras. Therefore, the accuracy can be improved by thorough calibrations in which calibration parameters would be identified differently for respective contact regions. By doing so, each fabricated sensor needs its own calibration of cameras, even though the design is similar. In this research, our proposed method does not require any calibration, thus accommodating unlikeliness in the fabrication process. One can consider using our method with calibrated parameters to increase the accuracy of the sensing operation. Last but not least, while the contact depth accuracy considerably depends on regions of the skin, this problem is not seen in the context of contact localization. In fact, as presented in Section V-E, the variation of localization errors is not significant when evaluated on a wide range of contact regions.

Overall, in the context of sim2real transfer for large-scale ViTac sensing, to our knowledge, the obtained results in this article set the benchmark for further development, and the reconstruction errors are within an acceptable range compared to previous work [11], [12].

D. Sim2Real Transferability of Contact Detection

This section examines whether the performance of contact detection learned from virtual data, specifically that are inferred from TacNet, can be transferred into the real data domain with the help of the R2S-GN model. We initially examine a suitable contact detection threshold ϵ_c^* that would maximize the detection capability using the virtual image dataset. The selection was conducted based on the analysis of the precision–recall tradeoff [39] of the touch classifier over a finite range of the



(a)

Input type	Precision	Recall	Accuracy
sim (virtual)	1.00	1.00	1.00
real (w/o R2S-GN)	0.95	1.00	0.95
real (via R2S-GN)	1.00	1.00	1.00

(b)

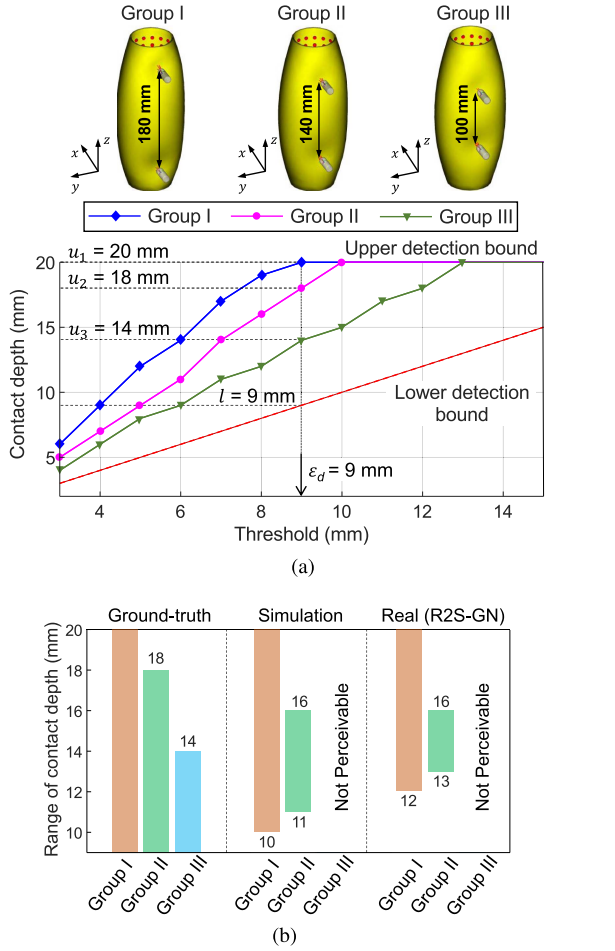
Fig. 11. Evaluation of contact sensing task. (a) Precision–recall tradeoff of the touch classifier evaluated on the virtual images. (b) Contact detection accuracy. The accuracy of contact classifier on different types of input images, with the decision threshold of 0.6.

decision threshold (ϵ_c). Based on the precision–recall plot [see Fig. 11(a)], it is reasonable to use a contact threshold value of 0.6 mm for sim2real evaluation, which maximizes contact sensing performance with 100% recall and precision.

The accuracy of the contact detection evaluated using test simulation dataset and corresponding real images is presented in Fig. 11(b). All the pure real images capturing the nondeformed skin are mistakenly classified as contact events (95% precision). However, the result [see Fig. 11(b)] reveals that this sim2real problem could be addressed by the intervention of the R2S-GN. In fact, real images passing through the R2S-GN model allow successful transfer of the threshold learned from the simulation to the domain of real images, in which the best precision and recalled values are retained (i.e., 100%).

E. Sim2Real Transferability of Double-Contact Localization

This subsection examines the accuracy and transferability of contact localization in the scenario of double contact. Three contact groups (groups I, II, and III) distinguished by the vertical distance between the two random contact points [180, 140, and 100 mm, respectively, as shown in Fig. 12(a)] were tested. For each group, based on our proposed localization method (see Section V-E), we determined from SOFA simulation data ($\mathbf{D}_{\text{FEM}}$) at what range of contact depth, the two separated contact areas were recognized with respect to the threshold ϵ_d [see (14)]. Group I showed the largest detection range followed by group II, and detection range increased with increase in threshold increases [see Fig. 12(a)]. Furthermore, Fig. 12(b) compares these results with estimated displacement data ($\mathbf{D}_{\text{estimated}}$) inferred from virtual tactile images (second column) and real tactile images



(a)

(b)

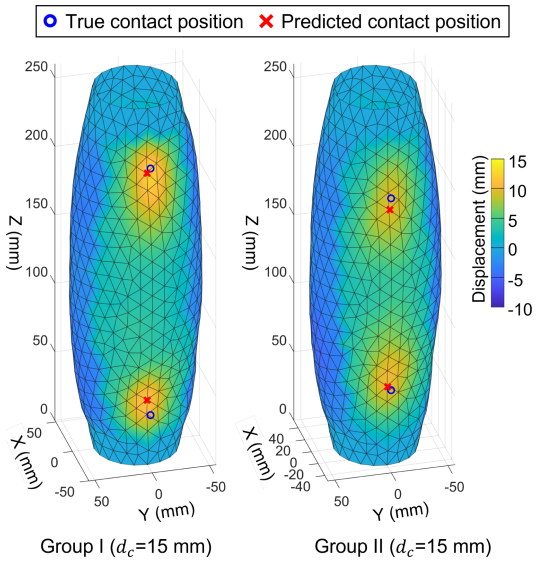
Fig. 12. Study of the sim2real transferability of two-point contact localization. (a) Range of contact depth permitting successful two-point detection versus decision threshold (ϵ_d), evaluated on SOFA ground-truth data ($\mathbf{D}_{\text{FEM}}$). (b) Sim2real comparison of contact depth range inside which the two-point contacts can be discriminated versus contact groups. ($\epsilon_d = 9$ mm).

(third column) at $\epsilon_d = 9$ mm, which yields the highest two-point detection accuracy. Except for Group I, which is still considered acceptable in both cases, the detection range of Group II was significantly down, while contacts among Group III (two-point distance is relatively close) were detected as one large contact area. The error came from the fact that our method utilized the displacement of nodes located not only in the actual contact sites but also in the regions surrounding them or the contact regions. That causes a narrow two-point detectable range of contact depth when ϵ_d is small [see Fig. 12(a)]. Other reason is that the occlusion is easier to occur when two points are close together. If two points share the same height or bias from a vertical direction, such situations are anticipated less critical than the tested cases due to the fact that every horizontal cross section of TacLink skin is parallel with image planes, so we could expect a clear vision of the contact areas (less occlusion).

Fig. 13(a) shows averaged localization errors between estimated and actual in-touch positions using simulation and transformed real tactile images for Groups I and II, while Fig. 13(b) visualizes the localization task in action at $d_c = 15$ mm. Overall,

Contact Point	Localization error (mm)			
	Group I		Group II	
	sim	real	sim	real
$\mathbf{x}_1^c$	$6.81 \pm 0.$	7.19 ± 1.06	$1.49 \pm 0.$	4.86 ± 4.64
$\mathbf{x}_2^c$	$6.81 \pm 0.$	7.18 ± 1.05	$1.49 \pm 0.$	7.10 ± 4.64

(a)



(b)

Fig. 13. Evaluation of two-point contact localization accuracy. (a) Double-point contact localization accuracy of simulation and transformed real dataset. (b) Demonstration for double-contact localization with transformed real samples for Group I and II ($d_c = 15$ mm).

the obtained results revealed the feasibility of sim2real transfer for multipoint contact localization. However, some sim2real gaps remain due to the fact that TacNet was generally trained with a poor double-touch database, and the R2S-GN did not possess prior relevant knowledge and was not powerful enough to handle such complex interactions.

VI. Applications of Large-Scale Tactile Sensing

This section describes two trial cases, including nonprehensile manipulation and haptic interface, for TacLink and its transferred tactile information of multipoint contact depth and contact location. In order to perform the tasks, a large-scale TacLink sensor utilized as a robot link (forearm) was attached to the elbow joint of a three-DoF custom-built robot arm [see Fig. 14(a)]. Note that spatial data defined in this section are all referenced to the fixed space frame {s} of the robot base, excluding those specified by left superscripted indices.

A. Nonprehensile Manipulation by Whole-Arm Pushing

This section showcases how a three-DoF robot arm could push an object toward a goal facilitated by an attached TacLink sensing device providing the function of contact location. For simplicity, we restricted the pushing task to be performed on a $\hat{y}_s\hat{z}_s$ plane of frame {s}. To perform the task, we employed simple proportional controllers that obtain feedback of the 3-D position of a pushed object $\mathbf{x}_{\text{object}} \in \mathbb{R}^3$ determined from *contact* made with TacLink, used to compute the desired spatial velocity ${}^c\mathcal{V}_d \in \mathbb{R}^6$ with reference to the contact frame {c} to guide the object toward the preset goal. The {c} frame, represented by rotation matrix $\mathbf{R}_c$, was defined to have the origin at the contact location $\mathbf{x}^c$ [retrieved from (18)], the $\hat{y}_c$ and $\hat{z}_c$ axes pointing along the outward normal of the contact plane, and along the z-axis of the TacLink frame (see Fig. 5), while the $\hat{x}_c$-axis complements the others by the right-hand rule. Given the object position ($\mathbf{x}_{\text{object}} \equiv \mathbf{x}^c$), and goal location $\mathbf{x}_{\text{goal}} \in \mathbb{R}^3$, the pushing task was controlled such that the pushing direction $\mathbf{n}_{\text{push}}$ was perpendicular to the contact plane ($\hat{y}_c \equiv \mathbf{n}_{\text{push}}$)

$$\mathbf{n}_{\text{push}} := \frac{\mathbf{x}_{\text{goal}} - \mathbf{x}_{\text{object}}}{\|\mathbf{x}_{\text{goal}} - \mathbf{x}_{\text{object}}\|}. \tag{20}$$

Therefore, the required angular velocity $\boldsymbol{\omega}_d \in \mathbb{R}^3$ to achieve the desired pushing direction can be devised as

$$\boldsymbol{\omega}_d = k_\omega \hat{\boldsymbol{\omega}}_d \bar{\theta} \tag{21}$$

where $k_\omega > 0$ is a proportional gain of angular velocity, and $\hat{\boldsymbol{\omega}}_d\bar{\theta} \in \mathbb{R}^3$ denotes the exponential coordinates of a rotation matrix $\bar{\mathbf{R}} := \text{Rot}(\hat{\boldsymbol{\omega}}_d, \bar{\theta}) = \mathbf{R}_{\text{push}}\mathbf{R}_c^{\text{T}} \in SO(3)$, where $\mathbf{R}_{\text{push}} := [\hat{x}_s,\ \mathbf{n}_{\text{push}},\ \hat{x}_s \times \mathbf{n}_{\text{push}}]$, that rotates the contact frame {c} toward the pushing direction. In addition, since the contacted object would deliberately be pushed along $\mathbf{n}_{\text{push}}$, the commanded linear velocity can be derived as

$$\mathbf{v}_d = k_v(\mathbf{x}_{\text{goal}} - \mathbf{x}_{\text{object}}) \tag{22}$$

On top of that, respecting a typical safe human–robot interaction scenario, we imposed a condition on the proposed pushing control to halt the robot motion in the event an unplanned contact (external contact) occurred, i.e., other than with the target pushed object. Hence, assuming that there always exists one contact with the target object, we have

$${}^c\mathcal{V}_d = \begin{cases} [\mathbf{0}]_{6\times 1}, & \text{if } L \geq 2 \\ \mathbf{R}_c^{\text{T}}[\boldsymbol{\omega}_d,\ \mathbf{v}_d]^{\text{T}}, & \text{otherwise} \end{cases}. \tag{23}$$

Finally, the desired twist ${}^c\mathcal{V}_d$ was mapped to commanded joint velocity $\dot{\boldsymbol{\theta}} \in \mathbb{R}^3$ through Jacobian ${}^c\mathbf{J} \in \mathbb{R}^{6\times 3}$ at the contact point.

The results of the described contact-based pushing experiment are shown in Fig. 14, wherein the goal position was set as $\mathbf{x}_{\text{goal}} = [-0.01,\ -0.17,\ 0.73]^{\text{T}}$, and the proportional gains were experimentally tuned as $k_v = 0.12$ and $k_w = 0.35$. During the pushing trial, the primary contact with the object (i.e., a water-filled bottle) maintained a relatively stable contact intensity of around 14 mm, except for when an unplanned contact occurred [see Fig. 14(b)]. Once a human suddenly touched the TacLink, triggering a secondary (external) contact phase, all robot motions halted, which resulted in a goal error (defined as $\|\mathbf{x}_{\text{goal}} - \mathbf{x}_{\text{object}}\|$), and the time-dependent object position ($\mathbf{x}_{\text{object}}$) remained unchanged [see Fig. 14(b) and (c)]. Over the course of time, the pushed object gradually reached the preset goal, which took around 60 s for the entire process. However, there remained a small degree of settling error along the z-direction, resulting in more or less 0.05-m goal error. This might be addressed by incorporating the integral term to the

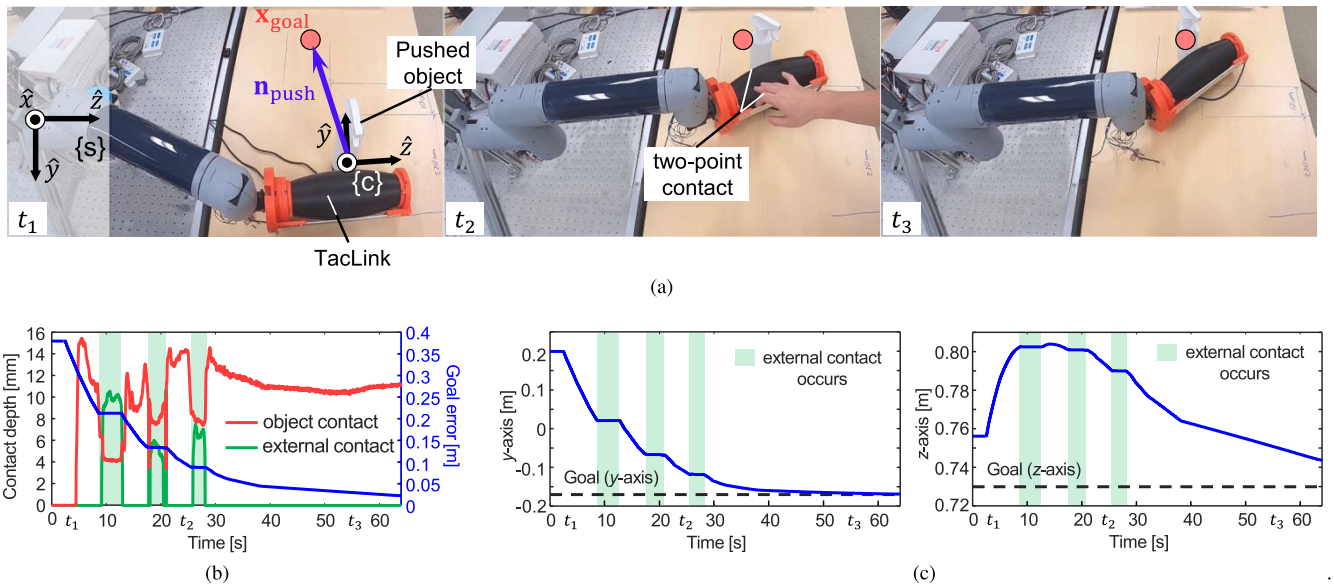


Fig. 14. Experiment of contact-based object pushing. An object, whose position is identified through contact with the TacLink, is guided to a goal location $\mathbf{x}_{\text{goal}} = [-0.01, -0.17, 0.73]^{\mathrm{T}}$ on a yz plane of a table via pushing. When unexpected contacts (external contacts) occurred, the robot motion was temporarily halted and then resumed after the external contact broke. The observations of the external contacts are green shaded. The demonstration can be found in the video at https://youtu.be/NN2u8YBLITY. (a) Video stills of contact-based object pushing experiment with the possible occurrence of two-point contact. (b) Time log of contact depth and positional error of the contacted object w.r.t the goal location. (c) Time log of the contacted object position measured along y and z axes of the space frame {s}. Goal lines indicate the positional references.

aforementioned proportional control law and requires further improvement in a more sophisticated control framework. More demonstrations of different goal locations are available in the supplementary video.

B. Haptic Interface for Motion Guidance

This section highlights the utilization of TacLink as a haptic interface device for intuitively guiding the motion of the robot arm [see Fig. 15(a)], where we strategically mapped tactile actions, including single/multipoint push and stroke into a desired robot twist ${}^{b}\mathcal{V}_d \in \mathbb{R}^6$. For the single-point push actions happening at contact location $\mathbf{x}^c$ [see (18)], the estimated contact depth $d(\mathbf{x}^c)$ [see (19)] and the normal direction $\mathbf{n}(\mathbf{x}^c) := \mathbf{N}_{i^*}$ [see (13)] are encoded to the spatial velocity on the $\hat{x}_b\hat{y}_b$ plane of the end-effector frame {b} as

$$[v_x,\ v_y,\ v_z]^{\mathrm{T}} = k_d^v d(\mathbf{x}^c)\mathbf{n}(\mathbf{x}^c) \tag{24}$$

where k_d^v is a constant to appropriately scale the resulting linear velocity. In addition, we employed distinguishable two-point contact as an interface for instructing rotational motion, wherein a virtual pivot point ${}^{b}\mathbf{r}_c$ was placed at the center of TacLink. Thus, the rotational motion around axes of {b} frame can be defined as

$$[w_x,\ w_y,\ w_z]^{\mathrm{T}} = k_d^v[{}^{b}\mathbf{r}_c \times d(\mathbf{x}_1^c)\mathbf{n}(\mathbf{x}_1^c) + {}^{b}\mathbf{r}_c \times d(\mathbf{x}_2^c)\mathbf{n}(\mathbf{x}_2^c)]. \tag{25}$$

Since, for simplicity, the push direction was restricted to the normal of a contact plane, we neglected the rotation/twist around the z-axis ($w_z = 0$), and the linear motion ($v_z = 0$) as well. However, the linear velocity along the z-direction could be induced by detecting the stroke action (SA). For robust stroke detection, we introduced a fixed time window $T_w = W.\Delta t$ (where W is the window size), during which the possible sliding motion on the skin surface is evaluated at a determined interval Δt. Therefore, at each time step Δt in the time window T_w, we measured the distance between the current contact position $\mathbf{x}^c_{k\Delta t}$ and the previous one $\mathbf{x}^c_{(k-1)\Delta t}$ as

$$\Delta x_k = \| \mathbf{x}^c_{k\Delta t} - \mathbf{x}^c_{(k-1)\Delta t} \| \quad \forall k \in \{1, 2, \ldots, W\}. \tag{26}$$

Now, let us denote $\mathcal{X} = \{\Delta x_k\}$; $|\mathcal{X}|$ is the number of its elements, and $\mathcal{K} = \{\Delta x_k \mid \Delta x_k \geq \epsilon_s\}, \forall k \in \{1, 2, \ldots, W\}$, where ϵ_s is a distance threshold to assure that the stroke is not misclassified due to the inaccuracy of contact localization [see Fig. 13(a)]. Hence, the SA along the z-axis can be recognized as

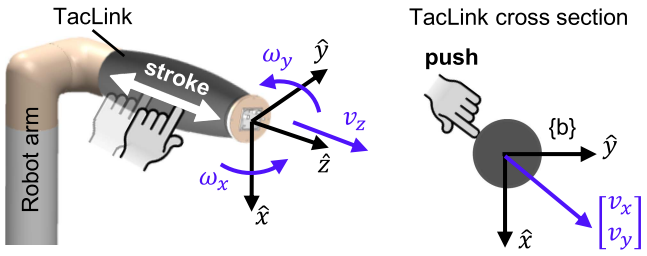
$$\text{SA} = \begin{cases} 1, & \text{if } |\mathcal{K}| \geq \eta|\mathcal{X}| \\ 0, & \text{otherwise (classified as push action)} \end{cases} \tag{27}$$

where η is a classification ratio experimentally set as 0.3. When a stroke occurs, from $t \geq T_w$, the linear velocity along the z-axis can be encoded as

$$v_z = \text{sgn}(x^{c,z}_{t+\Delta t} - x^{c,z}_t) k_d^{\omega} \frac{\left\| \mathbf{x}^c_{t+\Delta t} - \mathbf{x}^c_t \right\|}{\Delta t} \tag{28}$$

where $x^{c,z}_{t+\Delta t}$ and $x^{c,z}_t$ are the z-coordinates of $\mathbf{x}^c_{t+\Delta t}$ and $\mathbf{x}^c_t$, respectively, and k_d^{ω} is a constant scale of the angular velocity. Finally, the resultant desired twist ${}^{b}\mathcal{V}_d = [v_x,\ v_y,\ v_z,\ w_x,\ w_y,\ 0]^{\mathrm{T}}$ was mapped to commanded joint velocity $\dot{\boldsymbol{\theta}}$ through Jacobian ${}^{b}\mathbf{J}$ at the end-effector.

We validated the proposed interface scheme in several experiments, including various contact actions and scenarios. The control parameters are summarized in Table II. In the first showcase

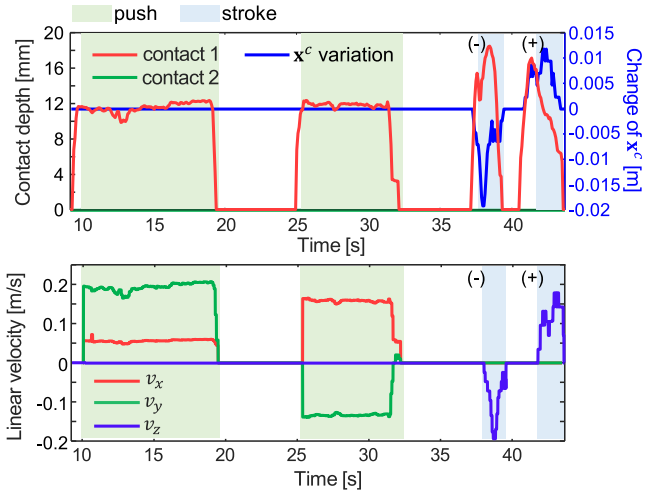


(a)

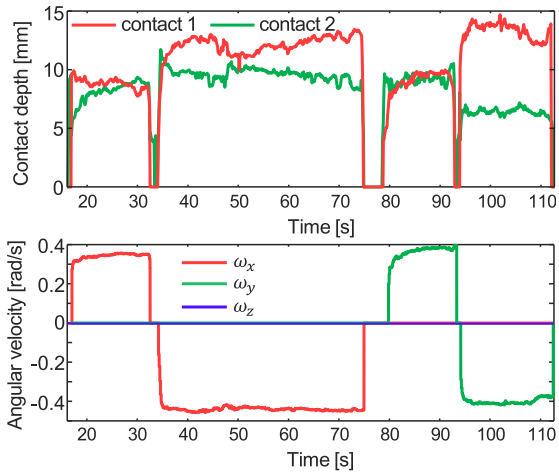
Action	Time [s]	Contact location $\mathbf{x}^c$ [mm]					
		x_1	y_1	z_1	x_2	y_2	z_2
Push	28.00	−41	34	−127	-	-	-
Stroke	37.00	−22	−48	−154	-	-	-
Stroke	37.06	−20	−48	−174	-	-	-
2-point	25.00	−2	−45	−224	3	45	−49
2-point	85.00	−42	−3	−196	34	2	−64

The contact position is expressed in the {b} frame.

(b)



(c)



(d)

Fig. 15. Motion guidance using TacLink as haptic interface. (a) Conceptual illustration for motion guidance. (b) Estimated contact locations in the experiments of different contact actions. The time of two-point actions is referred to (d), while that of the other ones respecting to (c). (c) Time log of contact depth and resulted robot linear velocity with respect to push and stroke contact action (shaded green and blue, respectively). (d) Time log of contact depth and robot angular velocity resulted from two-point contact action at different contact locations.

TABLE II
CONTROL PARAMETERS FOR PUSHING AND MOTION CONTROLLERS

Parameter		Value	Unit
Proportional gain (angular velocity)	k_ω	0.35	s^{-1}
Proportional gain (linear velocity)	k_v	0.12	s^{-1}
Goal location	$\mathbf{x}_{\text{goal}}$	$[-0.01,\ -0.17,\ 0.73]^{\mathrm{T}}$	m
Virtual pivot point	${}^b\mathbf{r}_c$	$[0,\ 0,\ -0.13]^{\mathrm{T}}$	m
Linear velocity scale	k_d^v	1.20	s^{-1}
Angular velocity scale	k_d^ω	0.15	s^{-1}
Stroke distance threshold	ϵ_s	8	mm

of *push* and *stroke* actions, the two actions could be distinguished by distinct patterns of the contact depth [see Fig. 15(c)]. In fact, a stroke, performed by a human digit, yielded relatively sharp changes in the contact depth profile, while stable intensity was observed with a push action. The linear velocity along the z-axis, resulting from the SA, was linearly scaled with the rate of contact positions over the duration $\Delta t = 0.03$ s, while the robot motion along the other two axes was triggered by the push action with the speed and direction depending on the contact depth and location [see Fig. 15(b)], respectively. Note that since the push/stroke classification requires the window size $W = 8$ to execute, the robot's response would be delayed for at least 0.24 s, just as the time window T_w. In addition, the delay time might increase due to the misclassification between the single- and two-point contact scenarios. While this problem could be addressed by more advanced classification algorithms (e.g., machine learning techniques), it can positively enable users to feel safer in the human–robot interactive tasks.

Furthermore, in the two-point contact condition [see Fig. 15(d)] and its specific pairs of contact positions [see Fig. 15(b)], the robot rotated around either $\hat{x}_b$ or $\hat{y}_b$. This showcase of haptic interface for motion guidance based on our large-scale tactile device is expected to provide initial hints for more action-based human–robot interaction strategies. The demonstration of robot motion guidance and other applications of TacLink on safety control can be found in the video.[4]

VII. DISCUSSION

A. Skin Geometry Affects Scalability and Extensibility

The showcase of SimTacLS in this article is based on the large-scale TacLink sensor, whose shape and size are unique

[4][Online]. Available: https://youtu.be/NN2u8YBLITY

and much different from the typical ViTac sensors reported in the literature. It is paramount that the working principle of SimTacLS does not depend on skin size or shape. Within the scope of this article mainly focusing on the applicability of SimTacLS to large-scale sensors, an in-depth discussion of the previous argument is essential.

Theoretically, skin geometry poses no restriction to the SOFA simulation tool as long as the material properties and boundary conditions of the skin are accurately given. Similarly, the Gazebo module is not dependent on skin geometry but on light conditions and on extrinsic and intrinsic parameters of the camera system. We posit that if the skin geometry allows the visibility of visual cues in tactile images at each tactile interaction phase, then SimTacLS is applicable. For example, owing to the nature of TacLink skin where markers are vertically aligned [see Fig. 2(a)], markers in the middle region are very sensitive to occlusion, especially during the contact phase. An adverse effect of such phenomenon on sensing performance was certified in [11] in the case of cylindrical skin and was expected to be more detrimental in the case of a barrel-shaped body. However, existing ViTac skins on bodies with flat or curved designs possess markers that are horizontally distributed all over the image plane. As a result, the effect of missing visual cues is minimized. This discussion strengthens our hypothesis that SimTacLS is promising for ViTac systems of various sizes and shapes.

On the other hand, regarding markers, it is obvious that their morphology and density determine the resolution and accuracy of obtained tactile information. Nonetheless, it is still questionable whether marker distribution or even marker design affects the overall performance of TacNet. Since SimTacLS is designed toward a unified platform for acquiring data and training online, it is worth investigating the morphological design of markers and their related meshing in SimTacLS to accelerate the process without compromising sensor accuracy. While the data acquisition process of the SimTacLS platform has been implemented offline, it is required to accelerate the computation process as much as possible as technology allows such as by multireading or GPU-based computing to meet potential requirements of real-time applications.

Regarding the bandwidth of the proposed sensing system, it depends on the required processing time and mechanical properties of the soft skin. Since trained networks are compact in size, real-time processing is not a problem as demonstrated in our previous analytical method [11]; therefore, the sampling rate is dependent on the frame rate of the camera. In this research, we could implement 120-Hz sampling rate on 120-frames/s cameras on TacLink. Therefore, the bandwidth of the sensor can be justified mostly based on the mechanical properties of the soft skin. For the TacLink, the stiffness of the skin can also be varied by the inner pressure value; thus, it is expected that the bandwidth of the sensor could be implemented online, given pretraining in SimTacLS.

B. Open Problems

1) Force Detection: In this article, SimTacLS was exploited to assess skin deformation resulting from contact occurring on the whole-body because such information can reveal features of tactile perception (contact location, size of contact area, vibration, etc.) at large scale. Human mechanoreceptors cannot convey in detail how much force is acting on skin. In addition, large-scale sensing is usually aimed at human–robot interactions rather than task-based ones, where information of force is deemed redundant. On top of that, toward a simulation framework for interactive robotics systems, TacNet was designed to be easily adaptable for different physical attributes, especially contact forces, other than the prediction of nodal displacements (skin deformation). In the future, for the physical formulation of interactive control problems, we aim to replace the current output signals of TacNet with nodal forces (which can be extracted from SOFA-based simulation), from which multicontact forces and locations at a large-scale skin can be effectively inferred. In fact, contact force information [$\boldsymbol{\lambda}$ in (1)] modeled from the SOFA kernel could be targeted to train TacNet models in which the same proposed sensing methods can be applied to extract high-level perception. Note that this process only requires the additional collection of contact forces and the pretraining of a TacNet model, but without any further change in the proposed pipeline.

2) Two-Point Touch Discrimination: As shown in Section V-E, TacLink could best detect two contact points (aligned vertically) separated by a distance of 140 mm, which yields an acceptable sensing behavior for a whole-arm ViTac device with a very soft skin. Related to the touch acuity of humans, large body parts, such as arms or torso, encode low spatial acuity of around 45 mm, while the two-point touch threshold of fingers is about 2–3 mm [42]. In fact, the two-point touch threshold of the present sensing device may be adjusted by varying the skin morphology, such as changing the skin material (e.g., stiffening) or increasing the air pressure in the enclosed skin. We may expect a shorter detectable two-point distance as the skin becomes stiffer, which results in the reduction of the number of deflected nodes under the same acting force [expressed through (3)]. In addition to a mechanical solution for adaptable sensing behavior, it is possible to enhance two-point spatial acuity by utilizing contact forces, which are represented by the Lagrange multipliers $\boldsymbol{\lambda}$ [see (1)], as a source for CRL algorithm rather than nodal displacements. Thereby, with the contact forces inferred by TacNet (trained on the force labels obtained from SOFA kernel), contact regions would be narrowed down to contain only the nodes in physical contact with the external environment.

3) Applications: In this article, we attempted to showcase the use of the TacLink device in task-based interactions, including object pushing, motion guidance, and contact detection/reaction, as highlighted in Section VI, by which we argue that these tasks are infeasible to achieve by existing small-scale tactile sensors. These preliminary demonstrations are also expected to lay the groundwork for more sophisticated tasks based on tactile sensing at large area, such as haptic exploration/manipulation in cluttered environments [43] and robot learning by demonstration. Last but not least, the provided tactile information could be integrated with proven high-level controls for other robot systems, such as mobile robots that are beyond the robot manipulator presented in this article.

C. Novelty

1) Large-Scale Tactile Sensing Problems: Even though previous works, such as TACTO [14] and Tactile Gym [15], could perform simulation on ViTac sensors in bodies of different shapes (e.g., hemisphere, cylinder, or flat), the feasibility of applying these simulators to large-scale sensors remains questionable. First, the operation of GelSight-like sensors strongly depends on the gel layer and the reflective-light work environment, which pose challenges in setting up suitable lighting conditions over a large area. Second, since external objects often come into contact with the large-scale TacLink in a direction perpendicular to the optical axis of a camera, there is a high possibility that two different contact locations may each or in combination yield imperceptible depth-based images. Hence, virtual depth-based images might provide insufficient and accurate tactile information in the context of large-scale ViTac soft sensors. Because of these possible problems as tactile devices scaled up, we strongly argue that a more realistic simulation platform with high-fidelity soft body interaction and realistic marker rendering, such as SimTacLS, is essential and worth investigating for accurate tactile sensing at a large scale.

2) Task Transferring Schemes: As reported in [15], Tactile Gym trained tactile-driven tasks, such as edge/surface following, object rolling, and so on, with the input of depth imprint images through reinforcement learning frameworks, which coupled tactile sensing with task performance. Thus, training a main task in the coupled manner may require setting up a new environment or retraining from scratch as tasks are newly defined (due to training losses defined differently on a task basis). In contrast, SimTacLS decoupled sensing problems from the desired end-task performance; thus, transferred tactile information was utilized as tactile feedback for control tasks, but not tactile images. By doing this, we could focus on integrating the transferred tactile information into potential or novel tactile-driven tasks.

VIII. Conclusion

In this article, we presented a pipeline named SimTacLS for simulation and training of a ViTac sensor at large area, taking into account compliant contact mechanics of the skin and actual showcases in robotics. The pipeline offers rich tactile information, particularly skin deformation, to learn sensing skills for a large-scale TacLink device. We demonstrated that a tactile neural network (TacNet), learned from the obtained simulation dataset, could trigger high-level tactile perception (i.e., contact detection and localization) with potential to benefit robotics tasks. In comparison with other tactile sensors (of different sensing principles), our system offers large-area sensing with a simple setup and least influence on the mechanical properties of the soft skin (no embedded sensing elements). Meanwhile, the proposed system requires large amount of data for training, which may increase the implementation time and cost. On top of that, the pipeline has possibilities for transferable learning of robotics tasks in virtual environments and leaves room for the scalability of a broader range of ViTac devices of diverse shapes and sizes. In the future, more elaborations on the application of the proposed system will be tackled to bring in a holistic approach for the implementation of large-area tactile sensing-based robotic scenarios.

Acknowledgment

The authors would like to thank Shotaro Nakayama and Nhat Dinh Minh Le for valuable advice at the initial stage and helping hands-on experiment setup. The authors would also like to thank Dr. Hugo Talbot for his recommendation on the construction of SOFA simulation and Dr. David Price for his proofreading of this article.

References

[1] R. S. Dahiya, G. Metta, M. Valle, and G. Sandini, "Tactile sensing—From humans to humanoids," *IEEE Trans. Robot.*, vol. 26, no. 1, pp. 1–20, Feb. 2010.

[2] S.-H. Hyon, J. G. Hale, and G. Cheng, "Full-body compliant human–humanoid interaction: Balancing in the presence of unknown external forces," *IEEE Trans. Robot.*, vol. 23, no. 5, pp. 884–898, Oct. 2007.

[3] A. Schmitz, P. Maiolino, M. Maggiali, L. Natale, G. Cannata, and G. Metta, "Methods and technologies for the implementation of large-scale robot tactile sensors," *IEEE Trans. Robot.*, vol. 27, no. 3, pp. 389–400, Jun. 2011.

[4] T.-H.-L. Le, P. Maiolino, F. Mastrogiovanni, and G. Cannata, "Skinning a robot: Design methodologies for large-scale robot skin," *IEEE Robot. Autom. Mag.*, vol. 23, no. 4, pp. 150–159, Dec. 2016.

[5] K. Park, H. Yuk, M. Yang, J. Cho, H. Lee, and J. Kim, "A biomimetic elastomeric robot skin using electrical impedance and acoustic tomography for tactile sensing," *Sci. Robot.*, vol. 7, no. 67, 2022, Art. no. eabm7187.

[6] K. Kamiyama, H. Kajimoto, N. Kawakami, and S. Tachi, "Evaluation of a vision-based tactile sensor," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2004, vol. 2, pp. 1542–1547.

[7] U. H. Shah, R. Muthusamy, D. Gan, Y. Zweiri, and L. Seneviratne, "On the design and development of vision-based tactile sensors," *J. Intell. Robot. Syst.*, vol. 102, no. 4, pp. 1–27, 2021.

[8] S. Zhang et al., "Hardware technology of vision-based tactile sensor: A review," *IEEE Sens. J.*, vol. 22, no. 22, pp. 21410–21427, Nov. 2022.

[9] B. Fang, X. Long, F. Sun, H. Liu, S. Zhang, and C. Fang, "Tactile-based fabric defect detection using convolutional neural network with attention mechanism," *IEEE Trans. Instrum. Meas.*, vol. 71, 2022, Art. no. 5011309.

[10] N. F. Lepora, Y. Lin, B. Money-Coomes, and J. Lloyd, "DigiTac: A DIGIT-TacTip hybrid tactile sensor for comparing low-cost high-resolution robot touch," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 9382–9388, Oct. 2022.

[11] L. Van Duong and V. A. Ho, "Large-scale vision-based tactile sensing for robot links: Design, modeling, and evaluation," *IEEE Trans. Robot.*, vol. 37, no. 2, pp. 390–403, Apr. 2021.

[12] V. A. Ho and S. Nakayama, "IoTouch: Whole-body tactile sensing technology toward the tele-touch," *Adv. Robot.*, vol. 35, no. 11, pp. 685–696, 2021.

[13] S. Yoshigi, J. Wang, S. Nakayama, and V. A. Ho, "Deep learning-based whole-arm soft tactile sensation," in *Proc. 3rd IEEE Int. Conf. Soft Robot.*, 2020, pp. 132–137.

[14] 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 Robot. Autom. Lett.*, vol. 7, no. 2, pp. 3930–3937, Apr. 2022.

[15] A. Church, J. Lloyd, and N. F. Lepora, "Tactile sim-to-real policy transfer via real-to-sim image translation," in *Proc. 5th Annu. Conf. Robot Learn.*, 2022, pp. 1645–1654.

[16] M. K. Johnson and E. H. Adelson, "Retrographic sensing for the measurement of surface texture and shape," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 1070–1077.

[17] B. Ward-Cherrier et al., "The TacTip family: Soft optical tactile sensors with 3D-printed biomimetic morphologies," *Soft Robot.*, vol. 5, no. 2, pp. 216–227, 2018.

[18] D. F. Gomes, P. Paoletti, and S. Luo, "Generation of GelSight tactile images for Sim2Real learning," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 4177–4184, Apr. 2021.

[19] A. Agarwal, T. Man, and W. Yuan, "Simulation of vision-based tactile sensors using physics based rendering," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2021, pp. 1–7.

[20] A. Padmanabha, F. Ebert, S. Tian, R. Calandra, C. Finn, and S. Levine, "OmniTact: A multi-directional high-resolution touch sensor," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 618–624.
[21] M. Lambeta et al., "DIGIT: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation," *IEEE Robot. Autom. Lett.*, vol. 5, no. 3, pp. 3838–3845, Jul. 2020.
[22] Z. Ding, N. F. Lepora, and E. Johns, "Sim-to-Real transfer for optical tactile sensing," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 1639–1645.
[23] C. Sferrazza, A. Wahlsten, C. Trueeb, and R. D'Andrea, "Ground truth force distribution for learning-based tactile sensing: A finite element approach," *IEEE Access*, vol. 7, pp. 173438–173449, 2019.
[24] C. Sferrazza and R. D'Andrea, "Sim-to-Real for high-resolution optical tactile sensing: From images to three-dimensional contact force distributions," *Soft Robot.*, vol. 9, no. 5, pp. 926–937, 2021.
[25] F. Zhang, J. Leitner, M. Milford, B. Upcroft, and P. Corke, "Towards vision-based deep reinforcement learning for robotic motion control," in *Proc. Australas. Conf. Robot. Autom.*, 2015, pp. 1–8.
[26] 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 *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 23–30.
[27] I. Goodfellow et al., "Generative adversarial nets," *Adv. Neural Inf. Process. Syst.*, vol. 27, 2014.
[28] N. H. Nguyen and V. A. Ho, "Tactile compensation for artificial whiskered sensor system under critical change in morphology," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 3381–3388, Apr. 2021.
[29] N. H. Nguyen and V. A. Ho, "Mechanics and morphological compensation strategy for trimmed soft whisker sensor," *Soft Robot.*, vol. 9, no. 1, pp. 135–153, 2022.
[30] J. Allard, H. Courtecuisse, and F. Faure, "Implicit FEM solver on GPU for interactive deformation simulation," in *GPU Computing Gems Jade Edition* (Applications of GPU Computing Series), W.-M. W. Hwu, Ed. Boston, MA, USA: Morgan Kaufmann, 2012, ch. 21, pp. 281–294.
[31] T. Liu, C. Zhao, Q. Li, and L. Zhang, "An efficient backward Euler time-integration method for nonlinear dynamic analysis of structures," *Comput. Struct.*, vol. 106–107, pp. 20–28, 2012.
[32] E. Coevoet et al., "Software toolkit for modeling, simulation, and control of soft robots," *Adv. Robot.*, vol. 31, no. 22, pp. 1208–1224, 2017.
[33] H. Courtecuisse, J. Allard, P. Kerfriden, S. P. Bordas, S. Cotin, and C. Duriez, "Real-time simulation of contact and cutting of heterogeneous soft-tissues," *Med. Image Anal.*, vol. 18, no. 2, pp. 394–410, 2014.
[34] J. A. González, K. Park, C. A. Felippa, and R. Abascal, "A formulation based on localized lagrange multipliers for BEM–FEM coupling in contact problems," *Comput. Methods Appl. Mech. Eng.*, vol. 197, no. 6, pp. 623–640, 2008.
[35] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham, Switzerland: Springer, 2015, pp. 234–241.
[36] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE / CVF Comput. Vis. Pattern Recognit. Conf. (CVPR)*, 2017.
[37] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Trans. Comput. Imag.*, vol. 3, no. 1, pp. 47–57, 2017.
[38] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[39] A. Gron, *Hands-On Machine Learning With Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 1st ed. Sebastopol, CA, USA: O'Reilly Media, Inc., 2017.
[40] S. Haddadin, A. De Luca, and A. Albu-Schäffer, "Robot collisions: A survey on detection, isolation, and identification," *IEEE Trans. Robot.*, vol. 33, no. 6, pp. 1292–1312, Dec. 2017.
[41] L. He, X. Ren, Q. Gao, X. Zhao, B. Yao, and Y. Chao, "The connected-component labeling problem: A review of state-of-the-art algorithms," *Pattern Recognit.*, vol. 70, pp. 25–43, 2017.
[42] S. Lederman and R. Klatzky, "Haptic perception: A tutorial," *Attention, Perception, Psychophys.*, vol. 71, no. 7, pp. 1439–1459, 2009.
[43] S. Zhong, N. Fazeli, and D. Berenson, "Soft tracking using contacts for cluttered objects to perform blind object retrieval," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 3507–3514, Apr. 2022.

Quan Khanh Luu (Student Member, IEEE) received the B.Eng. degree in mechatronics from the Ho Chi Minh City University of Technology, Ho Chi Minh City, Vietnam, in 2018, and the M.S. degree in robotics in 2021 from the Japan Advanced Institute of Science and Technology (JAIST), Nomi, Japan, where he is currently working toward the Ph.D. degree in robotics with Soft Haptics Labs.

He has worked as a Software Engineer with Bosch Corp., Ho Chi Minh City. His current research interests include robot learning and control, tactile sensing/perception, and applications for human–robot interaction.

Mr. Luu was honored as the Best Graduate Student for the outstanding academic performance during his master's degree.

Nhan Huu Nguyen (Member, IEEE) received the bachelor's degree from the Da Nang University of Science and Technology, Da Nang, Vietnam, in 2015, and the master's degree from the Ming Chi University of Science and Technology, New Taipei, Taiwan, in 2017, both in mechanical engineering, and the Ph.D. degree in robotics from the Japan Advanced Institute of Science and Technology (JAIST), Nomi, Japan, in 2022.

He is currently a Postdoctoral Researcher with the Soft Haptics Labs, JAIST. His research interests include exploiting complex physical interaction between deformable robots and the surrounding environment to enable novel functionalities, such as tactile sensing, or facilitate learning and controlling tasks.

Van Anh Ho (Senior Member, IEEE) received the bachelor's degree in electrical engineering from the Hanoi University of Science and Technology, Hanoi, Vietnam, in 2007, and the master's degree in mechanical engineering and the Ph.D. degree in robotics from Ritsumeikan University, Kyoto, Japan, in 2009 and 2012, respectively.

He completed the Japan Society for the Promotion of Science (JSPS) Postdoctoral Fellowship in 2013. He then joined Advanced Research Center Mitsubishi Electric Corp., Amagasaki, Japan, as a Research Scientist. From 2015 to 2017, he was an Assistant Professor with Ryukoku University, Kyoto, where he led a laboratory on soft haptics and soft modeling. Since 2017, he has been with the Japan Advanced Institute of Science and Technology, Nomi, Japan, for setting up a laboratory on soft robotics. His current research interests include soft robotics, soft haptic interaction, tactile sensing, grasping and manipulation, and bioinspired robots.

Dr. Ho was the recipient of the prestigious JSPS Research Fellowship for Young Scientist for his Ph.D. course (DC2) and Postdoctoral Fellowship. He was the recipient of the 2019 IEEE Nagoya Chapter Young Researcher Award, Best Paper Finalists at 2016 IEEE/SICE International Symposium on System Integration (SII) and 2020 IEEE-RAS International Conference on Soft Robotics (RoboSoft). He is Member of the Robotics Society of Japan. He is an Associate Editor for many international conferences, such as IEEE/RSJ International Conference on Intelligent Robots and Systems, SII, and RoboSoft, and for many journals, such as IEEE TRANSACTIONS FOR ROBOTICS, IEEE ROBOTICS AND AUTOMATION LETTERS, and *Advanced Robotics*. He is the General Co-Chair of IEEE/SICE SII2023, and General Chair of IEEE/SICE 2024.