Learning to Synthesize Volumetric Meshes from Vision-based Tactile Imprints

Zhu, Xinghao; Jain, Siddarth; Tomizuka, Masayoshi; van Baar, Jeroen

TR2022-055 June 03, 2022

Abstract

Vision-based tactile sensors typically utilize a de-formable elastomer and a camera mounted above to provide high-resolution image observations of contacts. Obtaining accu-rate volumetric meshes for the deformed elastomer can provide direct contact information and benefit robotic grasping and manipulation. This paper focuses on learning to synthesize the volumetric mesh of the elastomer based on the image imprints acquired from vision-based tactile sensors. Synthetic image-mesh pairs and real-world images are gathered from 3D finite element methods (FEM) and physical sensors, respectively. A graph neural network (GNN) is introduced to learn the image- to-mesh mappings with supervised learning. A self-supervised adaptation method and image augmentation techniques are proposed to transfer networks from simulation to reality, from primitive contacts to unseen contacts, and from one sensor to another. Using these learned and adapted networks, our proposed method can accurately reconstruct the deformation of the real-world tactile sensor elastomer in various domains, as indicated by the quantitative and qualitative results.

IEEE International Conference on Robotics and Automation (ICRA) 2022

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Learning to Synthesize Volumetric Meshes from Vision-based Tactile Imprints

Xinghao Zhu[1,2], Siddarth Jain[2], Masayoshi Tomizuka[1], and Jeroen van Baar[2]

***Abstract*— Vision-based tactile sensors typically utilize a deformable elastomer and a camera mounted above to provide high-resolution image observations of contacts. Obtaining accurate volumetric meshes for the deformed elastomer can provide direct contact information and benefit robotic grasping and manipulation. This paper focuses on learning to synthesize the volumetric mesh of the elastomer based on the image imprints acquired from vision-based tactile sensors. Synthetic image-mesh pairs and real-world images are gathered from 3D finite element methods (FEM) and physical sensors, respectively. A graph neural network (GNN) is introduced to learn the image-to-mesh mappings with supervised learning. A self-supervised adaptation method and image augmentation techniques are proposed to transfer networks from simulation to reality, from primitive contacts to unseen contacts, and from one sensor to another. Using these learned and adapted networks, our proposed method can accurately reconstruct the deformation of the real-world tactile sensor elastomer in various domains, as indicated by the quantitative and qualitative results.**

I. Introduction

Tactile is an essential sensing modality for humans when grasping and manipulating objects. Tactile sensors can provide direct information about contacts during robotic grasping and manipulation. Vision-based tactile sensors are variants among different designs for robotic tactile sensors [1]–[8]. These sensors use a camera to capture high spatial resolution images of the contact deformation of a piece of elastomeric gel with an opaque coating as the sensing surface, as shown in Fig. 1 (a) and (b).

Obtaining a mesh representation of the contact elastomer can advance the development of applications with vision-based tactile sensors, since meshes can provide accurate contact information. For instance, meshes of the elastomer have enabled in-hand object localization [9]–[11], vision-free manipulation [12]–[14], and contact profile reconstruction [3], [4], [15]–[17]. Also, meshes can be used for precise dynamics simulation [18]–[20] and future state estimation [21], [22].

Previous simulation studies [3], [4] for vision-based sensors focus on reconstructing the *surface mesh* by tracking markers on the sensor. This can provide the surface displacement fields of the elastomer. However, to better simulate the dynamics, a *volumetric mesh* is preferred [19]. Compared to the *surface mesh*, the *volumetric mesh* contains internal vertices and edges, thus can better encode the dynamics and estimate the contact profile with the Finite Element Method (FEM) [18], [23], [24]. Nevertheless, internal elements also challenge the reconstruction of the *volumetric mesh* due to additional dimensions. This paper addresses that challenge and proposes a method to directly predict the *volumetric mesh* from images using vision-based tactile sensors, such as the GelSlim [4], in a sim-to-real setting. Moreover, our approach does not rely on fiducial sensor markers to synthesize a *volumetric mesh.*

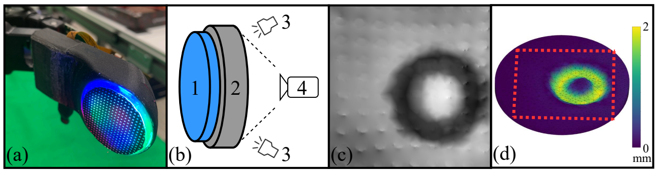

Fig. 1: **(a)** the GelSlim visual-tactile sensor, **(b)** the construction of the sensor, with the elastomer (1), the transparent lens (2), the lights (3), and the camera (4). **(c)** a depth image observation obtained from the sensor, and **(d)** the corresponding reconstructed volumetric mesh with our method. The red rectangle denotes the camera's view range, and the color represents the displacement level.

We first employ 3D FEM simulations of the GelSlim sensor's elastomer to collect image-mesh data pairs. The FEM simulations compute volumetric deformation fields for the elastomer with arbitrary contacts. The depth image observation is then rendered with synthetic cameras. The contact experiments are also executed in the real world with physical GelSlim. However, real-world contacts only provide images, since ground-truth meshes are unprocurable. We then learn mappings from real-world images to mesh deformations (as shown in Fig. 1 (c) and (d)) by leveraging supervised pre-training and self-supervised adaptations. Specifically, we learn an image-to-mesh projection in latent space with synthetic data pairs.

Sim-to-real approaches have to overcome the distribution differences between the two domains, that is the sim-to-real gap. We propose data augmentation of the synthetic images together with a self-supervised adaptation method on real-world images to address this gap. The adaptation uses a differentiable renderer to project the network output into images and minimize the difference between projected and input images. We demonstrate that this adaptation can transfer networks for sim-to-real, seen contact objects to novel contact objects, and between different GelSlim sensor instances. In this paper our goal is to introduce the synthesis of volumetric meshes from tactile imprints, and will address applications with our approach in the future work.

[1] Mechanical Systems Control Lab, UC Berkeley, Berkeley, CA, USA. {zhuxh,tomizuka}@berkeley.edu

[2] Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA {sjain,jeroen}@merl.com

Our work makes the following contributions:

- We provide a FEM model for GelSlim tactile sensors with a GPU-based simulator and propose a method to calibrate the FEM model with physical GelSlim sensors.
- We collect contact datasets from synthetic and real-world contact experiments for GelSlim sensors.
- We present an image-to-mesh projection network to reconstruct the volumetric mesh of the elastomer without the need for fiducial sensor markers.
- We further propose a self-supervised adaptation method and image augmentation techniques to mitigate the domain shift of sensor readings.

The problem formulation and details of our method are described in Section III, followed by experimental results in Section IV, and a discussion in Section V. We discuss the related work in the next section.

II. Related Work

A variety of tactile sensing capabilities for robotic applications have been introduced recently [25], [26]. In this paper, we limit our discussion to vision-based sensors, such as GelSight [1], and GelSlim [2].

It is nontrivial to convert the acquired tactile sensory information to quantities relevant for performing robotic tasks, such as the grasping force. While data-driven approaches to tactile sensing are becoming more popular, collecting large real-world datasets is not feasible due to expense and potential for damage. A promising solution is to investigate sim-to-real methods for tactile sensing capabilities.

Typically tactile sensors rely on soft materials, such as elastomers, wherein the contact results in deformation of the material. For vision-based tactile sensors, simulation thus involves modeling both visual and deformation behavior. Visual output of the GelSight sensor was simulated in [27] using a depth camera in the Gazebo simulator [28]. The approach involves computing the heightmap of the elastomer from the depthmap, and approximating the internal illumination of the elastomer using a calibrated Phong's reflection model. The illumination stage approach is further extended in [29] by leveraging OpenGL [30]. In our approach, we do not only attempt to bring the simulated tactile image closer to the real world, but we also use a data-driven approach with self-supervised adaption to map from tactile images to volumetric meshes. Furthermore, our approach has the potential to help with the need for texture augmentation, important in real-world robotic applications.

Analytical modeling with FEM simulation can model contact dynamics [23]. This has been used in the analysis of the behavior of soft materials for tactile sensing under various conditions. A variety of simulation methods have been considered for different types of tactile sensors using the FEM [3], [10], [31], [32].

The elastomer of the GelSlim tactile sensor is modeled as a linear elastic material in [3], and the FEM is used to compute the stiffness matrix to approximate its external forces and displacements. Using the surface displacements, this matrix is then used to compute an estimate of the force distribution. Unlike [3], in our approach we don't require the use of fiducial tracking markers to determine the displacement of the elastomer. Furthermore, we estimate volumetric meshes directly, which is not explored in prior work. Our approach is akin to the FEM model of the SynTouch BioTac sensor [32], in that both learn latent representations for the simulated sensor deformations and the real-world output through self-supervision. In contrast to [32], we focus on the image to volumetric mesh projection for vision-based tactile sensors. Since image observations from the sensor have higher variance and are noisy, our problem is more challenging.

III. Methods

This section first introduces the problem statement and preliminaries. Next, the image-to-mesh projection and self-supervised adaptation methods are discussed. Finally, the datasets are described, including synthetic labeled data, real-world unlabeled data, and the data augmentation techniques.

A. Problem Statement and Preliminaries

This paper focuses on the problem of reconstructing an elastomer's volumetric mesh with image observations for vision-based tactile sensors. The non-injective projection (or mapping) from surface images to volumetric vertex positions makes this problem nontrivial. Some preliminaries are described below:

1) Image Observations: Visual tactile sensors typically contact objects with a silicone elastomer and use a camera to capture the deformation of the surface, as shown in Fig 1. The captured RGB image can be used to construct a depth map of the contact surface using shape from shading [4], [33]. It establishes a mapping from the RGB color to the surface normals with a marble of known dimension. During runtime, surface normals are retrieved and integrated into the depth map I. Compared to raw RGB images, depth maps contain 2.5D information and can better represent the geometry of the contact surface [34]. Moreover, depth maps are much easier to simulate using synthetic cameras and thus have less sim-to-real gap. Therefore, in this paper we use (128×128) depth maps I as the image observations.

2) Volumetric Meshes with FEM: The FEM is a mathematical tool to solve complex partial differential equations (PDEs) [23]. In the FEM, geometrical shapes are represented by volumetric meshes $\mathcal{M}$, which consist of 3D elements, such as tetrahedrons and hexahedrons. With high-resolution meshes and small computation steps, FEM can estimate the forward dynamics of soft bodies [18], [24].

This paper uses graphs to represent volumetric meshes. Specifically, volumetric meshes are defined as a set of vertices and edges, $\mathcal{M} = (\mathcal{V}, \mathcal{A})$, with n vertices in 3D Euclidean space, $\mathcal{V} \in \mathbb{R}^{n\times3}$. The adjacency matrix $\mathcal{A} \in \{0,1\}^{n\times n}$ represents the edges. If vertices i and j are connected by an edge, $\mathcal{A}_{ij} = 1$, and $\mathcal{A}_{ij} = 0$ otherwise.

B. Supervised Image-to-Mesh Projection

Our goal is to map an input depth map I to a volumetric mesh $\mathcal{M}$. Although depth maps provide geometrical information for the contact surface, the projection from surface

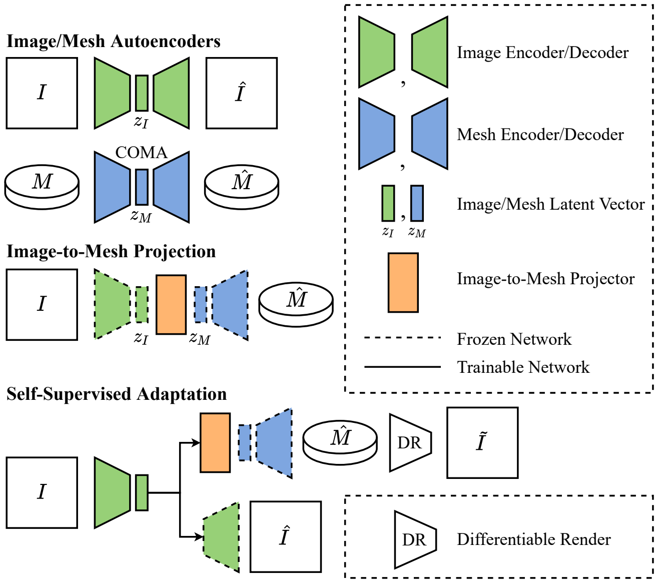

Fig. 2: Training structure. The image-to-mesh projection network is optimized with pre-trained autoencoders. The self-supervised adaptation transfers the projection network to various domains with a differentiable render.

images to volumetric vertex positions is not injective and is hard to analyze. Specifically, different displacements can generate the same surface observation. Thus, in this paper we assume a fixed mesh tessellation (i.e., $\mathcal{A}$ fixed) to enforce the injective mapping and use a neural network to learn the underlying projection $\hat{\mathcal{M}} = f_\theta(I)$, with θ being the parameters of the network.

The image-to-mesh projection is learned with latent representations. Compared to previous work [32], the image observations have higher variance and more noise. This paper introduces elaborate model designs, data augmentations, and self-supervised adaptations to resolve such difficulties.

Fig. 2 shows the training structure of the network. The image variational autoencoder (VAE) (in green) reconstructs depth maps I to $\hat{I}$ and is trained as a β-VAE:

$$\ell_I = \text{MSE}(I - \hat{I}) + \lambda_I \text{KL}(q(z_I|I) \parallel \mathcal{N}(0,1)) \qquad (1)$$

where q is the image encoder, λ_I is the weight for the KL divergence term, and z_I is the latent vector.

We adopt the convolutional mesh autoencoders (COMA) [35] for the volumetric mesh VAE (shown in blue). COMA uses spectral graph convolutional networks [36] to extract features and a hierarchical pooling operation to reduce vertices. The network is trained with:

$$\ell_M = \text{MSE}(\mathcal{M} - \hat{\mathcal{M}}) + \lambda_M \text{KL}(h(z_M|I) \parallel \mathcal{N}(0,1)) \qquad (2)$$

where h is the mesh encoder, λ_M is the KL loss weight, z_M is the latent vector, and the MSE is computed based on corresponding vertex positions $(\mathcal{V}, \hat{\mathcal{V}})$.

The latent projection model (shown in orange) is comprised of three fully connected layers. It is trained in a supervised manner with the encoder and decoder frozen. The details for the network are presented in Section IV-A. The latent dimensions and weights are chosen via hyperparameter search, which is discussed in Section IV-B.1.

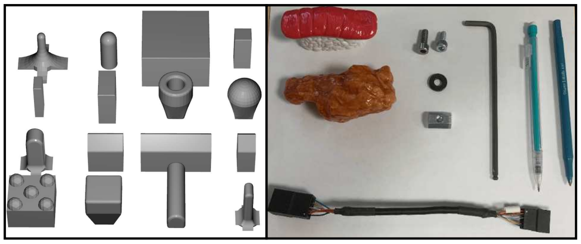

Fig. 3: *Left*: Primitive indenters in simulation. *Right*: Novel contact objects in the real world.

C. Self-Supervised Adaptation

When deploying the trained network to the real world, covariate shift problems may reduce the performance significantly [34]. Moreover, the real-world data only has depth maps $\{I_j\}$, the ground-truth volumetric meshes are not available, making it hard to fine-tune the network in a supervised manner. Thus, we propose a self-supervised adaptation framework (Fig. 2) to resolve the covariate shift.

Specifically, the reconstructed volumetric mesh $\hat{\mathcal{M}}$ is rendered to the image $\tilde{I}$ using a differentiable renderer, which allows gradients to propagate backward. In parallel, we use the pre-trained image VAE to reconstruct the input depth map $\hat{I}$. The image VAE works as a noise filter as suggested in [37]. In practice, removing the image VAE can lead to poor adaptation results, which is demonstrated in Section IV-C.1. The network is adapted using the mesh decoder with frozen weights, to minimize the loss:

$$\ell_{adapt} = \text{MSE}(\tilde{I} - \hat{I}) \qquad (3)$$

D. Datasets

Labeled synthetic data $\{(I_i, \mathcal{M}_i)\}$ and unlabeled real-world data $\{(I_j)\}$ are required to train the image-to-mesh projection and adapt the network among different domains.

1) Synthetic Data: Labeled image-mesh pairs $\{(I_i, \mathcal{M}_i)\}$ for $i \in [1, ..., N]$ can be simulated using FEM and synthetic cameras. In this work, FEM is performed using the GPU-based Isaac Gym [38]. Isaac Gym models the dynamics of deformable bodies using linear-elastic models and assumes isotropic Coulomb contacts. The results of the simulation are optimized to match the real-world deformation.

A FEM model for the GelSlim is created with a similar procedure as [32]. The elastomer pad is modeled as a cylinder with a 1.75cm radius and 0.3cm height. The volumetric mesh has 5,415 nodes and 23,801 edges. A rigid backplate is added to imitate the structure of the physical GelSlim, Fig 1(b). To generate labeled data pairs, 16 primitive indenters (Fig. 3–Left) are utilized to interact with the elastomer at randomized positions and rotations. The primitive shapes contain a variety of complexity, texture, and geometry to reflect daily household objects.

The Isaac Gym simulator collects vertex positions $\mathcal{M}$ at each contact trajectory. The depth map I is then rendered based on the mesh $\mathcal{M}$ with a synthetic camera. This paper

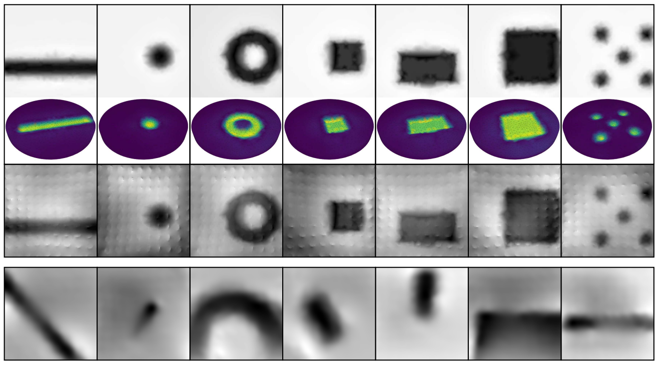

Fig. 4: Data samples. *Top*: Raw synthetic depth observations, corresponding ground-truth meshes, and augmented synthetic depth observations. *Bottom*: Real-world depth observations for sample indenters.

uses an orthographic camera with a ±1.75cm view range, which aligns with the specifications of the physical GelSlim. To optimize the FEM model in Isaac Gym, this paper reuses the calibration data in III-A.1, contact images of a marble of known dimensions. From the depth map I, the contact position can be accurately estimated by finding the maximum displacement point. The contact trajectory can then be reproduced in the simulation, which yields a deformed mesh $\mathcal{M}$. Then, a depth image $\tilde{I}$ is rendered based on the simulated mesh $\mathcal{M}$. The elastic modulus E, Poisson's ratio ν, and surface friction μ are designated as free parameters in the simulator. A cross-entropy search strategy is used to find the best parameters:

$$E, \nu, \mu = \arg\min_{E,\nu,\mu} \left\| I - \tilde{I} \right\|$$

The optimal values for E, ν, μ are 145MPa, 0.32, 0.94, respectively. Fig. 4 shows examples of synthetic data pairs with the calibrated FEM model.

2) Real-World Data: Real-world datasets $\{I_j\}$ are obtained with physical GelSlim sensors and various indenters (Fig. 4). Primitive indenters are 3D printed and interaction with the sensor is randomized. Besides primitive shapes, several household and industrial objects are used as a novel set (Fig. 3–Right). The novel set represents common objects that the GelSlim will work with. Moreover, we use two GelSlim sensors to collect real-world data.

3) Image Augmentations: As shown in Fig. 4, the appearance of synthetic images is quite different from that of real-world depth maps. The depth reconstruction process for the physical GelSlim introduces significant noise into the image, enlarging the sim-to-real gap. To enhance the performance in the real world, this paper injects Perlin noise and adds a real-world reference noise image into the synthetic images [34]. The Perlin noise provides a realistic gradient for the image and imitates the real-world camera noise. The reference image provides sensor-specific noise. Fig. 4 provides examples of the noised images.

In total, 1.28M unique labeled image-mesh pairs were obtained from the simulator, and 1,651 real-world images were obtained for 2 GelSlim sensors with 19 indenters.

TABLE I: Experiments with synthetic data pairs. The root-mean-square error (RMSE, in *cm*) is measured between the ground-truth vertex positions $\mathcal{M}$ and predicted vertex positions $\hat{\mathcal{M}}$. (a) the results with different dimensions of latent space, (b) the results with different loss weights.

(a)

$d(z_M)$ $d(z_I)$	64	128	256
64	0.221	0.152	0.200
128	0.232	**0.141**	0.192
256	0.210	0.167	0.189

(b)

λ_M λ_I	0	200	400	800
0	0.141	0.073	0.124	0.150
100	0.082	**0.012**	0.025	0.037
200	0.094	0.035	0.031	0.046

IV. Experiments & Results

In this section, we present the network details, experiments for supervised image-to-mesh projection, self-supervised adaptation, and a comparative evaluation with a baseline.

A. Network Details

As described in Section III-B and III-C, we use an image VAE, a mesh VAE, and a latent projection module. In the image VAE, the encoder includes five downsampling layers with feature sizes 32, 64, 128, 256, 512 and two fully connected layers with 128 neurons each. In the volumetric mesh VAE, the encoder consists of four Chebyshev convolutional filters [36] with feature sizes 16, 16, 16, 32 and an output fully connected layer with 128 neurons. Each Chebyshev convolution is down-sampled by a factor of four. The image and mesh decoder are symmetric with the encoders. The latent projection module has three fully connected layers with 256, 512, and 256 neurons. All networks use the Adam optimizer with a learning rate of $1e-3$ and decay of 0.99.

B. Supervised Projection

Our proposed supervised image-to-mesh projection depends on several hyperparameters. In this section we empirically estimate these. Furthermore, we pre-train the VAEs prior to training the image-to-mesh projection. We evaluate the pre-training by comparing with training the image-to-mesh projection directly from scratch.

The results reported here use a $80/20$ split on the synthetic dataset for training and validation. Each model was trained for 300 epochs. We report the mean validation root-mean-square-error (RMSE) for the projected meshes.

1) Latent Dimensions: We compare the image-to-mesh projection results for a 64, 128, and 256-dimensional latent space for each VAE, shown in Table I (a). The 128-dimensional latent space for both VAEs gives the best results.

2) Loss Weights: We also compared the effectiveness of different values for λ_I, λ_M, from eqs. (1) and (2), shown in Table I (b). We can see that the variational encoding, i.e., $\lambda_I > 0$, $\lambda_M > 0$, significantly improves the performance of latent projection, with best performance for $\lambda_I = 100$, $\lambda_M =$

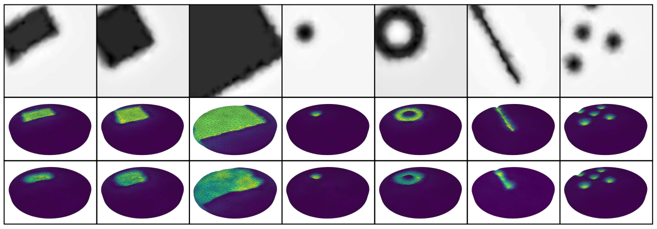

Fig. 5: Image-to-mesh projection results with synthetic data. *First row*: Input depth observations. *Second row*: Corresponding ground-truth mesh. *Third row*: Reconstructed volumetric mesh with our approach.

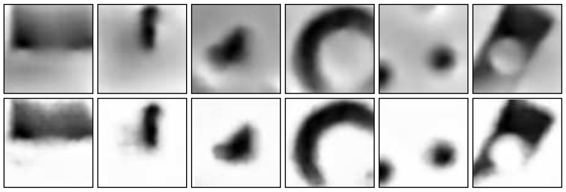

Fig. 6: Reconstruction results for the image VAE with real-world images. *First row*: Real-world image observations. *Second row*: Reconstructed image with pre-trained VAE. The image VAE can effectively remove visual noises for both primitive and novel contacts.

200. This suggests that the KL divergence term enforces a more meaningful latent distribution compared to a vanilla autoencoder. Fig. 5 shows a batch of projection results using the best performing model.

3) Pre-training: Given the best performing model, we investigate the usefulness of the VAE pre-training. We trained the image-to-mesh network from scratch with variational encoding. The training and validation errors were 0.009*cm* and 0.085*cm*, respectively. This suggests that the network overfits without the pre-training, which aligns with the findings presented in [32].

C. Self-Supervised Adaptation

We propose a self-supervised adaptation method and synthetic data augmentations to resolve the covariate shift problem, as discussed in III-C and III-D.3. This section provides results and ablation studies for the proposed method. We show that neither adaptation nor augmentation can achieve the objective alone, and the image VAE improves the adaptation results. Finally, we demonstrate that the proposed methods can adapt networks from simulation to reality, from primitive to novel contacts, and from one sensor to another.

The adaptation is performed with the real-world dataset $\{I_j\}$, without ground-truth mesh availability. To evaluate the performance of the adaptations, we use the RMSE between $\hat{I}$ and $\tilde{I}$ as the evaluation metric, where $\hat{I}$ is the reconstructed input depth map via the pre-trained image VAE. As shown in Fig. 6, we can observe that the image VAE is robust in different domains and can effectively remove noise. Specifically, we tested the VAE on augmented synthetic images. Results show that the pre-trained image VAE can reconstruct the clean depth map with a RMSE of 0.07*cm*.

TABLE II: Experiments with real-world data. The root-mean-square error (RMSE) is measured between reconstructed images $\tilde{I}$ and rendered images $\hat{I}$. (a) ablation studies for adaptation, data augmentation, and VAE filtering. (b) domain adaptation results.

(a)

	RMSE (*cm*)
Adapt + Aug.	**0.12**
No Aug. No Adapt	1.03
Only Aug.	0.57
Only Adapt	0.79
Adapt + Aug. w/o VAE	0.87

(b)

Source → Target	RMSE before/after Adaptation (*cm*)
Sim-Prim. → *Real-Prim*	0.57 → 0.12
Sim-Prim → *Real-Prim-2*	0.77 → 0.20
Real-Prim → *Real-Prim-2*	0.35 → 0.16
Real-Prim → *Real-Novel*	0.64 → 0.41
Sim-Prim → *Real-Novel*	1.30 → 0.62

Networks were trained or tuned on source domains and then adapted to target domains. The RMSEs were measured before and after the adaptation.

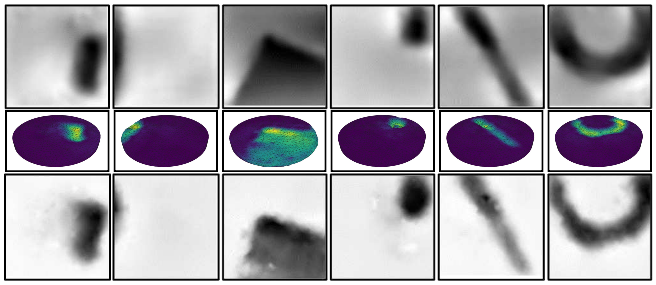

Fig. 7: Experiments with real-world primitive contact objects. *First row*: Input depth observations. *Second row*: Reconstructed volumetric meshes. *Third row*: Rendered depth images from reconstructed meshes.

1) Ablation Studies: We compare the effects of the adaptation model, synthetic data augmentations, and image VAE filtering. The results are listed in Table II (a). As the table shows, the data augmentation and self-supervised adaptation both contribute to resolving the sim-to-real gap. We observe that using only adaptation, or only augmentation, results in lower performance. The reason for higher performance when both are combined is two-fold. On one hand, the data augmentation enlarges the distribution of the synthetic dataset, which causes the real-world data to be within distribution (or close to). On the other hand, the adaptation model transfers the network from the simulated distribution to the real-world distribution, ensuring invariant feature encodings. Table II (a) also shows that the VAE filter improves adaptation performance. It removes visual noises in real-world data and stabilizes the adaptation process. A batch of qualitative reconstruction examples is shown in Fig 7.

2) Domain Adaptations: Sections III-D.1 and III-D.2 introduce various data domains, including simulated data with primitive contact objects (*Sim-Prim*), real-world data with primitive contact objects (*Real-Prim*), real-world data with

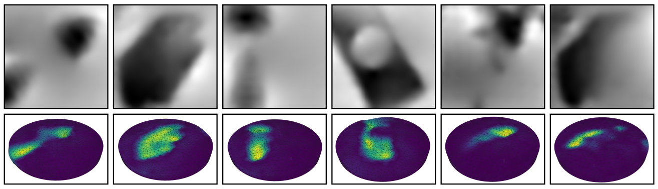

Fig. 8: Experiments with real-world novel contact objects. *First row*: Input depth observations. *Second row*: Reconstructed volumetric mesh from the network.

novel contact objects (*Real-Novel*), and real-world primitive data with a second GelSlim sensor (*Real-Prim-2*).

While we showed the performance of the *Sim-Prim* → *Real-Prim* experiment above, Table II (b) and Fig. 8 show the transfer results among other domains. The networks were first pre-trained or fine-tuned on source domains and then adapted to target domains. Experiments *Sim-Prim* → *Real-Prim*, *Sim-Prim* → *Real-Prim-2*, and *Real-Prim* → *Real-Prim-2* were executed with the same primitive shapes. The adaptation improves performance in all cases. For experiment *Real-Prim* → *Real-Novel*, acquisition was done with the real sensor, but adaptation now is for primitive to novel shapes. From Table II (b) we see that while the performance improves, improvement is less compared to the prior experiments. For the final experiment *Sim-Prim* → *Real-Novel* transfer is both from sim-to-real, as well as from primitive to novel shapes, and thus is hardest. Again, adaptation significantly improves performance and predicted deformations were visually accurate (see Fig. 8). The results suggest that the proposed adaptation method can effectively improve the performance of the network under both visual noise and shape differences.

Overall performance for experiment *Sim-Prim* → *Real-Novel* is less compared to the other experiments. The covariate shifts for visual noise and shape differences are not correlated, and adaptation for each separately performs better compared to adaptation for both. Further optimizing performance for both in a self-supervised manner is a challenging topic for future work.

D. Baseline Comparisons

For regression from image observations to mesh deformations, two methods were evaluated: 1) our proposed method, denoted as *Volumetric Mesh*, and 2) a surface reconstruction baseline [3], [4], denoted as *Surface Mesh*. The latter uses tracking markers to determine the movement of the elastomer surface. Note that the *Surface Mesh* method does not estimate the volumetric mesh directly, but rather gives a sparse surface deformation field for each contact.

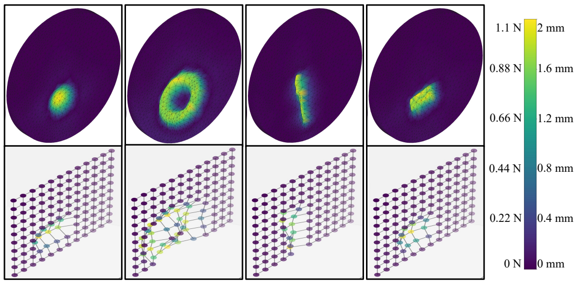

Fig. 9: *First row*: Reconstructed meshes and estimated contact forces with the proposed approach, *Volumetric Mesh*. *Second row*: Comparison with baseline method, *Surface Mesh* [4].

Fig. 9 shows the reconstructed meshes with both methods. Interestingly, the computation takes 0.02 *sec*. for the *Volumetric Mesh* synthesis with our proposed approach versus 0.04 *sec*. for the *Surface Mesh* method (potentially due to the requirement of marker detection). Fig. 9 shows correspondences between the *Volumetric Mesh* and the *Surface Mesh* on the elastomer surface. In addition, we also conducted contact force estimations of the GelSlim based on the predicted meshes. An inverse FEM was used to compute the contact force with a linear-elastic model [4]. Compared to the *Surface Mesh* method, our method constructed more plausible and denser force distributions with the volumetric FEM mesh (Fig. 9). For example, predictions around contact edges were more realistic and had higher resolution. Predicted force profiles were also smoother, which was due to the influence of internal vertices. We hypothesize that such denser force distributions obtained from our method may help improve policy learning for robotic manipulation tasks.

V. Discussion and Conclusion

This paper presents a framework to synthesize volumetric meshes of vision-based tactile sensor for novel contact interactions. Our work has several key contributions. First, we present a 3D FEM simulator for vision-based tactile sensors and a simulator calibration approach. Second, we generate a dataset for the GelSlim sensor with both simulated and real-world contacts using primitive and novel shapes. Third, we propose a label-free adaptation method and image augmentations for domain transfers; we show that this approach can effectively transfer networks to various visual and different shape scenarios. Lastly, our network efficiently reconstructs the volumetric mesh with depth images and precisely estimates the contact profiles of different shapes. Using these learned and adapted networks, our method can reconstruct the deformations of the elastomer for vision-based tactile sensors in various domains, as indicated by the quantitative and qualitative results.

The present work also has some limitations. First and foremost, although volumetric meshes can obtain dense force estimation and contact patch reconstruction, we do not explicitly demonstrate the application of tactile volumetric meshes on a robotic task. Instead, the focus of this work is on the synthesis of volumetric meshes from tactile imprints. Second, the current network cannot predict the dynamics of the elastomer, which may be prohibitive for performing model predictive control applications, as these require valid prediction of futures. In our future work, we will focus on addressing these limitations and develop volumetric mesh-based techniques for extensive robotic manipulation tasks, such as rope manipulation, peg-in-hole insertion, extrinsic contact estimation, and contact prediction.

References

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

[2] E. Donlon, S. Dong, M. Liu, J. Li, E. Adelson, and A. Rodriguez, "Gelslim: A high-resolution, compact, robust, and calibrated tactile-sensing finger," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

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

[4] I. Taylor, S. Dong, and A. Rodriguez, "Gelslim3.0: High-resolution measurement of shape, force and slip in a compact tactile-sensing finger," *ArXiv Preprint*, vol. abs/2103.12269, 2021.

[5] C. Matl, J. Koe, and R. Bajcsy, "Stretch: a soft to resistive elastic tactile hand," *arXiv preprint arXiv:2105.08154*, 2021.

[6] B. W. McInroe, C. L. Chen, K. Y. Goldberg, K. Y. Goldberg, R. Bajcsy, and R. S. Fearing, "Towards a soft fingertip with integrated sensing and actuation," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[7] M. Lambeta, P.-W. Chou, S. Tian, B. Yang, B. Maloon, V. R. Most, D. Stroud, R. Santos, A. Byagowi, G. Kammerer, *et al.*, "Digit: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 3838–3845, 2020.

[8] A. Padmanabha, F. Ebert, S. Tian, R. Calandra, C. Finn, and S. Levine, "Omnitact: A multi-directional high-resolution touch sensor," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[9] M. Bauza, O. Canal, and A. Rodriguez, "Tactile mapping and localization from high-resolution tactile imprints," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019.

[10] Y. S. Narang, K. V. Wyk, A. Mousavian, and D. Fox, "Interpreting and predicting tactile signals via a physics-based and data-driven framework," *ArXiv Preprint*, 2020.

[11] M. Bauza, E. Valls, B. Lim, T. Sechopoulos, and A. Rodriguez, "Tactile object pose estimation from the first touch with geometric contact rendering," *ArXiv Preprint*, 2020.

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

[13] F. R. Hogan, J. Ballester, S. Dong, and A. Rodriguez, "Tactile dexterity: Manipulation primitives with tactile feedback," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[14] Y. She, S. Wang, S. Dong, N. Sunil, A. Rodriguez, and E. Adelson, "Cable manipulation with a tactile-reactive gripper," in *Robotics: Science and Systems (RSS)*, 2020.

[15] N. F. Lepora, A. Church, C. de Kerckhove, R. Hadsell, and J. Lloyd, "From pixels to percepts: Highly robust edge perception and contour following using deep learning and an optical biomimetic tactile sensor," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2101–2107, 2019.

[16] D. Ma, S. Dong, and A. Rodriguez, "Extrinsic contact sensing with relative-motion tracking from distributed tactile measurements," *ArXiv Preprint*, 2021.

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

[18] T. Pfaff, M. Fortunato, A. Sanchez-Gonzalez, and P. W. Battaglia, "Learning mesh-based simulation with graph networks," in *International Conference on Learning Representations*, 2021.

[19] A. Sanchez-Gonzalez, N. Heess, J. T. Springenberg, J. Merel, M. Riedmiller, R. Hadsell, and P. Battaglia, "Graph networks as learnable physics engines for inference and control," *ArXiv Preprint*, 2018.

[20] A. Sanchez-Gonzalez, J. Godwin, T. Pfaff, R. Ying, J. Leskovec, and P. W. Battaglia, "Learning to simulate complex physics with graph networks," in *International Conference on Machine Learning*, 2020.

[21] F. d. A. Belbute-Peres, T. D. Economon, and J. Z. Kolter, "Combining Differentiable PDE Solvers and Graph Neural Networks for Fluid Flow Prediction," in *ICML*, 2020.

[22] Y. Li, T. Lin, K. Yi, D. Bear, D. L. Yamins, J. Wu, J. B. Tenenbaum, and A. Torralba, "Visual grounding of learned physical models," in *ICML*, 2020.

[23] M. A. Neto, A. Amaro, L. Roseiro, J. Cirne, and R. Leal, *Finite Element Method for 3D Solids*. Cham: Springer International Publishing, 2015, pp. 233–263.

[24] E. G. Thompson, *Introduction to the Finite Element Method: Theory, Programming and Applications*. Wiley Text Books, 2004.

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

[26] A. Yamaguchi and C. G. Atkeson, "Recent progress in tactile sensing and sensors for robotic manipulation: can we turn tactile sensing into vision?" *Advanced Robotics*, vol. 33, no. 14, pp. 661–673, 2019.

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

[28] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004.

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

[30] Khronos Group, "OpenGL," https://www.opengl.org/, 2021, [Online; accessed September-2021].

[31] 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. 173 438–173 449, 2019.

[32] Y. Narang, B. Sundaralingam, M. Macklin, A. Mousavian, and D. Fox, "Sim-to-real for robotic tactile sensing via physics-based simulation and learned latent projections," *Proceeding of the 2021 International Conference on Robotics and Automation (ICRA)*, 2021.

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

[34] X. Zhu, L. Sun, Y. Fan, and M. Tomizuka, "6-dof contrastive grasp proposal network," in *Proceedings of the 2021 International Conference on Robotics and Automation (ICRA)*, 2021.

[35] A. Ranjan, T. Bolkart, S. Sanyal, and M. J. Black, "Generating 3D faces using convolutional mesh autoencoders," in *European Conference on Computer Vision (ECCV)*, 2018.

[36] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in Neural Information Processing Systems*, 2016.

[37] R. Lopez, J. Regier, M. I. Jordan, and N. Yosef, "Information constraints on auto-encoding variational bayes," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2018, p. 6117–6128.

[38] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State, "Isaac gym: High performance gpu-based physics simulation for robot learning," *ArXiv Preprint*, 2021.