Taxim: An Example-based Simulation Model for GelSight Tactile Sensors

Zilin Si¹ and Wenzhen Yuan¹

Abstract-Simulation is widely used in robotics for system verification and large-scale data collection. However, simulating sensors, including tactile sensors, has been a long-standing challenge. In this paper, we propose Taxim, a realistic and high-speed simulation model for a vision-based tactile sensor, GelSight [1]. A GelSight sensor uses a piece of soft elastomer as the medium of contact and embeds optical structures to capture the deformation of the elastomer, which infers the geometry and forces applied at the contact surface. We propose an example-based method for simulating GelSight: we simulate the optical response to the deformation with a polynomial look-up table. This table maps the contact geometries to pixel intensity sampled by the embedded camera. In order to simulate the surface markers' motion that is caused by the surface stretch of the elastomer, we apply the linear displacement relationship and the superposition principle. The simulation model is calibrated with less than 100 data points from a real sensor. The example-based approach enables the model to easily migrate to other GelSight sensors or its variations. To the best of our knowledge, our simulation framework is the first to incorporate marker motion field simulation that derives from elastomer deformation together with the optical simulation, creating a comprehensive and computationally efficient tactile simulation framework. Experiments reveal that our optical simulation has the lowest pixel-wise intensity errors compared to prior work and can run online with CPU computing. Our code and supplementary materials are open-sourced at https://github.com/CMURoboTouch/Taxim.

I. INTRODUCTION

Simulation has been widely applied in robotics. It enables roboticists to quickly generate large amounts of realistic data, without costly equipment, manual labour, and the risk associated with real-world experiments. With growing interest in robot simulation, well-developed simulation frameworks such as Gazebo [2], PyBullet [3], MuJoCo [4], Drake [5], SOFA [6], NVIDIA Isaac Gym [7] have been widely used in the robotics community. They can simulate dynamic rigid-body, soft-body, vision and laser sensors with varying levels of accuracy and speed. However, none of them have integrated simulation of tactile sensing which form an irreplaceable part of robotic systems.

Recent advancements in vision-based tactile sensors, such as GelSight [8], [1], have made high-resolution tactile sensing available. These sensors use a piece of soft elastomer, or *gelpad*, as the contact medium for interacting with the environment. There is typically a printed marker array on the gelpad surface that moves as the surface stretches and is a good indicator of the contact forces and torques. The



Fig. 1: The GelSight outputs when it contacts objects with rich textures. With the ground-truth geometry [9], our simulation model generates images that are very similar to the real ones.

sensor utilizes optical components, including LEDs and an embedded camera to capture the illumination change caused by the change of light reflection on the galpad surface when the sensor contacts an external object, as shown in Fig 3. Simulating those vision-based tactile sensors, which contains modeling both the mechanical response of the soft gelpad and the optical response to the deformation, is challenging. There have been previous studies on simulating different components of vision-based tactile sensors separately. For instance, Ding et al. [10] built a physics soft body simulation for the TacTip [11] sensor to indicate pins' motion on the soft membrane; Agarwal et al. [12] and Gomes et al. [13] applied physics-based models for vision-based tactile optical simulation; whereas Wang et al. [14] integrated the optical simulation of tactile sensors with the physics simulation engine PyBullet. However, the work mentioned above lack the ability to simulate the intrinsic noise of the real sensors. They also demand heavy computation while are difficult to generalize to new sensors.

In this work, we propose *Taxim*, an example-based tactile simulation model that combines *optical simulation* and *marker motion field simulation*. The method overcomes the constraints of the previous simulation models in that it is computationally lightweight and generates very similar

¹Zilin Si and Wenzhen Yuan are with the Robotics Institute, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA, 15213, USA <zsi, wenzheny>@andrew.cmu.edu

signals to the real sensors in spite of the intrinsic noise of the sensors. The model contains two parts:

- 1) A polynomial table mapping function to simulate the *optical response* of a GelSight sensor by mapping geometries to pixel intensity in tactile images and an accumulation approach to simulate the shadow caused by the illumination.
- 2) A model to simulate the *marker motion field* using the linear displacement relationship and the superposition principle for the gelpad's elastic deformation.

Taxim is calibrated with less than 100 contact examples, so that it can easily migrate to other vision-based tactile sensors with similar designs as GelSight. To the best of our knowledge, Taxim is the first model that simulates all functions of vision-based tactile sensors, including the optical response for geometry measurement and marker motion field for force/torque measurement. Our simulation model can be integrated into robot simulation engines to provide a useful tool for tacile-based manipulation research.

II. RELATED WORK

A. Tactile Sensing Simulation

The majority of tactile sensors use a soft medium for contact where the measurement of its deformation can indicate the contact information. Therefore the sensor simulation has been mostly focused on simulating the elastomer deformation. Typically, elastic soft body simulation is modelled with finite element methods (FEM) [15], mass-spring model [16], particles [17] or learning methods [18]. To simulate tactile sensors that use soft medium, a traditional way is to build an approximation model of the soft bodies. Pezzementi et al. [19] and Moisio et al. [20] simulated the low dimensional tactile sensor signals with a point spread function model and a soft contact model with a full friction description respectively. However, they are not applicable to the high-resolution vision-based tactile sensors such as GelSight. Narang et al. [21] used a finite element model to simulate the BioTac sensor [22] and contact data in ANSYS. As an extension work, they simulated the deformation of BioTac in Issac gym [7] and projected this deformation to electronic signal with a generative learning framework in [23]. Sferrazza et al. [24], [25] built a synthetic dataset with a finite element model of the vision-based tactile sensors, trained a network to predict the 3D contact force distribution in simulate and realized sim-to-real transfer. In this work, instead of using an accurate finite element model with high-computing cost, we approximate the deformation of the soft medium on the GelSight sensor with pyramid Gaussian kernels which is efficient and also gives acceptable accuracy.

B. Optical Simulation for Tactile Sensors

For vision-based tactile sensors like GelSight, optical simulation is essential as it is used to measure geometries of the contacted objects. To simulate the optical system of GelSight, Gomes et al. [13] and Hogan et al. [26] used Phong's model to simulate the reflection and illumination. Agarwal et al. [12] applied ray tracing to simulate the light



Fig. 2: The pipeline of our proposed example-based simulation model.

paths within the sensor that form tactile images. Wang et al. [14] presented TACTO, an open-source simulator using pyrender to simulate DIGIT [27] sensors and bridged it to a physics simulator PyBullet. Compared to those physics-based methods, our method is data-driven, so that it is computationally efficient and can better simulate the intrinsic noise of the real sensors.

C. Marker Motion Field Simulation for Tactile Sensors

The movement of marker array on GelSight or other vision-based tactile sensors is caused by the planar stretch of the elastomer surface. They also make the key component of many vision-based tactile sensors such as TacTip [11]. In manipulation tasks such as slip detection [28] and grasping stability prediction [29], the marker motion serves as an essential feature. For TacTip, Ding et al. [10] simulated the dynamics of its soft membrane in Unity so as to extracted markers' motion. They evaluated the simulation on simto-real robot tasks. Church et al. [30] simulated the depth maps to represent the contact geometries instead of optical tactile images. They also used a Generative Adversarial Networks (GANs) to realize real-to-sim translation for TacTip sensors. Unlike the above work, we explicitly simulate the marker motion field by using FEM offline and applying the superposition principle [31] online. Our method does not require extensive training data but only a gelpad FEM model, and it approximates the marker motion field well with high accuracy and low computation cost.

III. METHODS

A. Overview

We construct and employ our simulation models for both optical response and marker motion of GelSight sensors. To simulate the sensor's optical response, we build a polynomial table to map the contact geometries to the image intensities and collect shadow masks to attach the shadow. Then we apply the superposition principle based on the loading displacement of each finite unit to simulate the markers' motion. Their combination replicates the contacting of objects on the tactile sensor. For both parts, we calibrate our simulator



Fig. 3: (a) Demo of the photometric stereo method: for a surface point p under the light l, the reflected light intensity captured by the camera is determined by the surface reflectance and the surface normal \vec{n}_p (b) The GelSight sensor [32] we aim to simulate and (c) its schematic diagrams of the optical structure.

with examples from a real sensor. We show the pipeline for building and applying the simulation model in Fig. 2.

B. Optical Simulation

We simulate the optical response of the GelSight sensor as a result of the contact geometry with a model using the examples-based photometric stereo [33] method. Photometric stereo uses the linear reflection function to derive the illumination of the object with the light sources and shape of the illuminated surface. Example-based photometric stereo does not require prior knowledge of lights sources but instead uses the imaging of the reference objects. We use a lookup table as the baseline and a polynomial table as our proposed method to map the contact shapes to image intensities.

a) Lookup Table Mapping: The gelpad has a homogeneous diffuse internal surface which makes the reflection function spatial-invariant. The linear reflection function used by photometric stereo is formalized as $I_p = \rho \mathbf{n_p} \sum \mathbf{l}$, where at a point p, the observed light intensity I_p caused by reflection is a product of the albedo ρ , the surface normal $\mathbf{n_p}$ and the light direction and intensity \mathbf{l} , as shown in Fig. 3 (a). The previous equation implies that the reflected light intensity I and the surface normal \mathbf{n} are linearly correlated. Alternatively, instead of solving the equation from the given lighting conditions, an intensity-shape lookup table can be built as follows

$$I = \sum_{l} a^{l} \mathbf{n} \tag{1}$$

where a^l is the coefficient of the light l, and can derived from a calibration process similar to [8], [34]. Here, we define the lights of the same color–any of red, green or blue–as one light source, even though they are contributed by multiple LEDs.

b) Polynomial Table Mapping: The linear lookup table works well with the point lights which are far from the object, whose directions are parallel and intensities are uniform for all the points on the illuminated surface. However, the LEDs in the GelSight are close to the sensor surface so that the emitted light is not strictly parallel and uniform. To compensate for the complicated lighting conditions, we introduce a non-linear model for the reflection as proposed



Fig. 4: Data collection setup (a, b, c) and data examples (d, e) to build the optical simulation model. The GelSight is placed on an XYR optical stage and an indenter with a certain shape object is mounted on a vertical linear stage for precisely indenting on the GelSight. We calibrate the polynomial table with less than 100 data points using a spherical indenter and collect shadow masks with 10 data points using a pin indenter.

in [35]. The reflection function can then be rewritten as

$$I = \sum_{l} f_{\mathbf{n}}^{l}(x, y) \tag{2}$$

where **n** is the normal vector representative of the surface shape and (x, y) is the 2D location on the image plane. From experiments, we found that in practice a second order polynomial function is sufficient to approximate the nonlinearity. Thus, the non-linear function is represented as:

$$f_{\mathbf{n}}^{l}(x,y) = \mathbf{w}_{\mathbf{n}}^{l}\mathbf{b}$$
(3)

where $\mathbf{w}_{\mathbf{n}}^{l}$ is a 6×1 vector that represents the parameters to model the polynomial table, and $\mathbf{b} = [x^{2}, y^{2}, xy, x, y, 1]^{T}$.

c) Calibration: Calibration entails fitting the parameters in the polynomial table from real data. Since these parameters vary for different sensors, this process has to be done per sensor. During calibration, we press a small ball with a known radius over the surface and manually locate contact areas in the tactile images as shown in Fig 4 (b) and (d). The surface normal at each point in the circular area can be easily calculated based on the ball's geometry. We discretize the 3D surface normal vectors to a 125×125 table with the magnitude and direction of the surface normal as the two dimensions. The parameters in polynomial table can then be solved via least squares with the set of intensity-shape-location pairs $(I_p, \mathbf{n}_p, x_p, y_p)$ from collected data. We fill invalid values in the table by interpolation.

d) Simulation: We simulate the visual outputs in three steps: collision detection, deformation approximation, and optical simulation. A collision is detected when an object comes in contact with the gelpad. From this contact, the local shape, represented as a height map, is constructed from the object's shape in the contact area, and gelpad's shape in non-contact area. Additionally, we need to simulate the soft body deformation from the height map. An approximation of soft body simulation is applied with pyramid Gaussian kernels. The shape in contact area is kept unchanged to maintain the fine textures and the boundaries between the contact and non-contact areas are smoothed using pyramid Gaussian kernels



Fig. 5: Shadow synthesis. (a) A unit shadow case observed under the lighting. (b) We approximate the object as the composition of unit pin case and then attach the shadow caused by each pin. (c), (d) and (e): We collect a set of shadow masks and synthesize the shadow around the contact area.

from large to small. From the height map, the normal vector for each point can be extracted and mapped to an intensity value with the calibrated polynomial table to synthesize the tactile images.

C. Shadow Simulation

Other than the illumination change that is modeled with photometric stereo method, the shadow is another factor causing the change of the pixel intensity in the tactile images. According to the design of the sensor, the shadows are caused by three groups of LED lights: red, green and blue lights. We simulate the shadow from those light sources respectively. We simplify shadow casting by collecting the "unit" shadow case, and then simulate the shadow by accumulating the shadows caused by each geometrical "unit". Since each light beam is traveling independently in the space, without considering inter-reflection, the shadow cast by them can be linearly accumulated.

A "unit" shadow is the shadow cast by a standing pin, as shown in Fig. 5 (a). For objects with different geometries, we consider them as the accumulation of "unit" shadows placed side by side with different heights, as illustrated in Fig. 5 (b). Therefore, given the tactile images with shadow cast by indenting a pin normally onto the gelpad with different depths as shown in Fig. 4 (c) and (e), we extract a set of shadow masks on three dominant directions caused by three light sources. For a general case, the shadow mask is attached for all three color channels and all points within the contact area if the neighbors are lower than that point.

D. Marker Motion Field Simulation

We simulate the markers' motion on the gelpad surface caused by the deformation of the soft gelpad from contacting. In this work, we consider the deformation under normal and shear loads. We employ the linear displacement relationship and superposition principle [31] to compose the deformation of the surface with loads on each finite unit of the contact surface. Although the markers are sparsely spread on the surface, we mesh the surface with dense nodes in simulation, track each node's motion and then locate the markers' motions. With the dense solution, the method can be applied to markers at any locations. Nodes in the gelpad surface mesh are classified into two categories: *active nodes* who come in contact with the object and are applied external forces and constrained by internal elastic forces; *passive nodes* who are in non-contacted area and only constrained by the internal elastic forces.

The linear displacement relationship assumes that any two nodes can influence each other in a linear way. By considering two nodes n_i and n_j with displacement in 3D as $\mathbf{u_i}$ and $\mathbf{u_j}$, the n_j can be passively influenced by n_i as $\mathbf{u_j} = T_{n_j}^{n_i} \mathbf{u_i}$, where $T_{n_j}^{n_i}$ is a 3×3 tensor representing the mutual influence. The superposition principle states that a node n_i 's displacement $\mathbf{u_i}$ is an aggregation of all active nodes' influence to it. Assume we have active nodes' $K = \{k_1, k_2, k_3, ..., k_m\}$, where $\mathbf{u_i^k}$ represent active nodes' initial displacement under external loads. Under the linear displacement relationship and superposition principle, any node n_i 's displacement $\mathbf{u_i}$ can be composed as

$$\mathbf{u}_{\mathbf{j}} = \sum_{i=1}^{m} T_{n_j}^{k_i} \mathbf{u}_{\mathbf{i}}^{\mathbf{k}}$$
(4)

However, before applying the superposition principle by using the active nodes' initial displacement $\mathbf{u}_{i}^{\mathbf{k}}$, we need to amend them to virtual displacements under the virtual loads because they not only are constrained by external loads but also influence each other. For instance, if all the active nodes' displacements are initialized such that they only move along the z direction *i.e.* $\mathbf{u}^{\mathbf{k}} = [0, 0, dz]$, the following equation holds:

$$\mathbf{u}_{\mathbf{j}}^{\mathbf{k}}[z] = \sum_{i=1}^{m} T_{k_j}^{k_i}[3,3] \tilde{\mathbf{u}}_{\mathbf{i}}^{\mathbf{k}}[z]$$
(5)

where $\mathbf{u}_{\mathbf{j}}^{\mathbf{k}}$ is the initialized displacement, and $\mathbf{u}_{\mathbf{j}}^{\mathbf{k}}[z]$ is its component along z-direction; $\tilde{\mathbf{u}}_{\mathbf{i}}^{\mathbf{k}}$ is the virtual displacement under the virtual load; $T_{k_j}^{k_i}[3,3]$ is the last element in the tensor $T_{k_j}^{k_i}$. Therefore, it is able to solve virtual displacements for active nodes by stacking all the equations as:

$$\mathbf{u}^{\mathbf{k}}[z] = M_{z}\tilde{\mathbf{u}}^{\mathbf{k}}[z]$$

$$\begin{bmatrix} \mathbf{u}_{1}^{\mathbf{k}}[z] \\ \mathbf{u}_{2}^{\mathbf{k}}[z] \\ \vdots \\ \mathbf{u}_{m}^{\mathbf{k}}[z] \end{bmatrix} = \begin{bmatrix} T_{1}^{1}[3,3] & T_{2}^{1}[3,3] & \dots & T_{m}^{1}[3,3] \\ T_{1}^{2}[3,3] & T_{2}^{2}[3,3] & \dots & T_{m}^{2}[3,3] \\ \vdots \\ T_{1}^{m}[3,3] & T_{2}^{m}[3,3] & \dots & T_{m}^{m}[3,3] \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{u}}_{1}^{\mathbf{k}}[z] \\ \tilde{\mathbf{u}}_{2}^{\mathbf{k}}[z] \\ \vdots \\ \tilde{\mathbf{u}}_{m}^{\mathbf{k}}[z] \end{bmatrix}$$
(6)
Then $\tilde{\mathbf{u}}^{\mathbf{k}}[z]$ is solved by matrix inversion as $\tilde{\mathbf{u}}^{\mathbf{k}}[z] =$

Then $\mathbf{u}[z]$ is solved by matrix inversion as $\mathbf{u}[z] = M_z^{-1} \mathbf{u}^{\mathbf{k}}[z].$

The x, y components of the active nodes' displacement can be amended using the same approach, but with T[1, 1]or T[2, 2] for the x or y directions respectively. Later, we apply the superposition principle to get the final resultant displacements for all nodes with

$$\mathbf{u}_{\mathbf{j}} = \sum_{i} T_{n_j}^{k_i} \tilde{\mathbf{u}}_{\mathbf{i}}^{\mathbf{k}} \tag{7}$$



Fig. 6: Elastic deformation calibration and simulation for the gelpad. We calibrate the deformation of gelpad under a unit pin with 0.5 mm diameter indenting in ANSYS to get the dense nodal displacement results (left). In simulation, we compose each node's displacement from elastic deformation in three steps: (a) initial displacement boundary conditions on active nodes, (b) active nodes' virtual displacements, and (c) resultant nodal displacements for both active nodes in contact area and passive nodes in non-contact area.

a) Calibration: The tensor $T_{n_i}^{k_i}$ depends on the gelpad's physical properties, and therefore can be measured in advance. The markers on the real gelpad are sparsely distributed which can not be used to generate dense meshes. Instead, we calibrate the arbitrary $T_{n_i}^{k_i}$ in a Finite Element Method (FEM) software ANSYS. In ANSYS, we generate the dense mesh of the gelpad and measure the deformation when there is a load on a unit node, as shown in Fig. 6 (left). We then use the measurement of the deformation to calibrate T. Since the markers are not printed on the top surface of the gelpad, we extract the second layer's mesh which is 0.5mm below the top surface from the simulated model as reference. To fully calibrate the 3×3 tensors, we simulate an active node's motion in z-direction only, a combination of z-direction and x-direction and a combination of z-direction and y-direction. Then we solve all the tensors $T_{n_i}^{k_i}$ using least squares from these three sets of unit case.

b) Simulation: We employ the marker motion simulation in three steps, by: 1) applying the initial displacements on the active nodes under the external loads, 2) getting active nodes' virtual displacements with the superposition principle, and then 3) calculating the resultant displacements at each node using the superposition principle with virtual displacements of active nodes. This process is demonstrated in Fig. 6 (a), (b), (c).

IV. EXPERIMENTS

We perform a set of experiments to evaluate the similarity between the simulated tactile data and that from real sensors.

A. Experiment Setup and Data Collection

To collect well-controlled contact data with a real GelSight sensor, We set up an optical platform, as shown in Fig. 4 (a). The GelSight is placed on a XYR stage, and an indenter is mounted on a vertical linear stage positioned above the GelSight. We manually control the contact location and depth by adjusting the stages. The XYR stage enables horizontal movement and the vertical stage adjusts the indenting depth. Both are with 0.01mm precision. We use a dome-shaped gelpad for both the real sensor and the simulated sensor.

We evaluate our simulation using objects with different shapes and textures. The objects are designed in Solidworks [36], output as mesh files for simulation (Fig. 7 (a))



Fig. 7: Dataset of objects designed in Solidworks (a) and 3D printed (b) for contact experiments. The objects are of different shapes. Their base sizes are either $10\text{mm} \times 10\text{mm}$ or $15\text{mm} \times 15\text{mm}$.

and 3D printed for collecting data from the real sensor for comparison (Fig. 7 (b)).

B. Optical Simulation

To calibrate the optical simulation model, we collect 50 data points on different locations of gelpad surface with a 4mm-diameter spherical indenter, as shown in Fig. 4 (b), (d); to calibrate the shadow simulation model, we collect 10 data points of different pressing depths with a 1mm-diameter pin indenter, as shown in Fig. 4 (c), (e). The calibration process is simple and easy to conduct manually without precise control of contact locations which can be accomplished within 1 hour. And it can be used till any components of the sensor are replaced or the sensor is broken.

We simulate the tactile images on the aforementioned dataset and compare our method with three other methods: the physics-based model [12], TACTO [14] and Phong's model [13] as shown in Fig. 8. We evaluate our method by comparing the simulated images with the real images in pixel-wise level, against the three methods mentioned above on four metrics: mean absolute error (L1), mean squared error (MSE), structural index similarity (SSIM) and peak singal-to-noise ratio (PSNR). The simulated images are cropped to size 400×400 around the indenting area to eliminate the background's effect. Also, due to the precision of the operation with the real sensor, the ground truth tactile images are not well aligned with the simulated images. So we manually align the images using GIMP [37]. The quantitative results are summarized in the Table. I. From the table, our method outperforms all the other methods.



Fig. 8: Optical simulation comparison among our method, TACTO [14], Phong [13] and physics [12] with the real data.

	L1 ↓	$\mathbf{MSE}\downarrow$	SSIM ↑	PSNR ↑
Tacto [14]	10.861	215.861	0.808	25.495
Phong's [13]	8.163	123.249	0.832	27.763
Physics [12]	7.409	90.623	0.759	28.687
Ours	5.565	58.358	0.882	30.974

TABLE I: Image similarity metrics between simulation and real data for optical simulation. We compare our method with methods from Physics-based model [12], Tacto [14] and Phong's model [13] on L1, MSE, SSIM and PSNR metrics. Our method performs the best on all the metrics.

Different indentation depths and locations Our optical simulation model works well for different indentation depths and locations. One example is shown in Fig. 9. From MSE errors, we can see errors increase when the indentation become farther from the center and deeper.

Fine texture simulation Our model can simulate the contact cases with fine-textured objects, as shown in Fig. 1.

Simulation on various sensors and objects Note that tactile images look different in Fig. 8, Fig. 9 Fig. 11 and Fig. 13. This is because we use 4 different GelSight sensors and manufactures lead to the difference. However, our model works well on all of them. We also apply our model on a DIGIT sensor [27] and the results are shown in Fig. 10. In addition, We test our model on various objects from the Google Scan dataset [38] and some results are shown in Fig. 11.

Speed test We test all the simulation techniques, mentioned above, on a AMD Ryzen Threadripper 2950X 16-Core Processor CPU. We input height maps with the size 480×640 , and output the simulated tactile images of the dataset. We then record the average running time of all the methods, as shown in the Table II. Our method is the most computationally lightweight on CPU and achieves the realtime data transferring speed from real sensors. However, Tacto [14] and Physics model [12] can be largely accelerated on GPUs but not considered here for evaluation. Our method can be potentially optimized for GPU computation as well and we will work on that for the next step.



Fig. 9: Optical simulation results with different indentation depths and locations. The locations differ over the gelpad surface while the depths differ as 0.5mm, 1.0mm, 1.5mm. The MSE error is shown below each pair.



Fig. 10: Optical simulation results (right) for a DIGIT sensor (left).

Speed	Ours w/o	Ours w/	Physics	Tacto	Phong's
	shadows	shadows	[12]	[14]	[13]
Frequency (fps)	18.1	9.6	0.1	1.9	3.8

TABLE II: Speed test for optical simulation on CPU. We compare our method with the Physics-based model [12], Tacto [14] and Phong's model [13]. Our method runs with the fastest speed.

C. Marker Motion Field Simulation

We evaluate the simulation results with two references: 1) the dense displacement map generated by the FEM simulation, and 2) the sparse displacement map collected from a real sensor. The contact cases are with objects in Fig. 7 under combinations of different normal loads and shear loads. The load displacement varies from 0.3 mm to 0.8 mm.

Comparison with FEM simulation As illustrated in the four sets of comparison from Fig. 12, the dense mesh vertices displacements on X, Y, Z are simulated from both the FEM (Fig. 12 (b) left) and our methods (Fig. 12 (b) right). The color red means negative displacement value and blue means positive values. The average interpolated pixel-wised L1 errors over the gelpad surface on dataset are 3.58×10^{-3} mm for X-axis, 3.32×10^{-3} mm for Y-axis, 5.43×10^{-3} mm for Z-axis, and 5.40×10^{-3} mm for XY (gelpad surface).

Comparison with both real data and FEM simulation Examples of results are shown in Fig. 13. The mean of marker motion's magnitude L1 errors on dataset is 1.00×10^{-2} mm between real & FEM, 1.02×10^{-2} mm between real & ours, and 3.96×10^{-3} mm between FEM & ours. We weight the marker motion's angular errors based on the



Fig. 11: Optical simulation results for objects from the Google Scan dataset [38]. We touch the objects with a GelSight mounted on a robot arm given certain contact locations. Most artifacts in the simulated images come from the coarse mesh files of the objects.



Fig. 12: The simulated marker motion field in a dense mesh in comparison with the FEM data. The heat maps show the marker motion field in X, Y and Z directions where positive Z is the direction of normal loads and X, Y are the direction of shear loads.

its magnitude because smaller marker motion is easier being affected by the system noise. The weighted mean of marker motion's angular L1 errors is 12.94° between real & FEM, 14.57° between real & ours, and 4.89° between FEM & ours. From the experimental results, the FEM model and our model match well, but there is still a gap from the simulation to the real gelpad soft body model. Three reasons observed from the experiments causing the errors are: 1) The gelpad is hand-manufactured and it is not perfectly matched with the FEM model in ANSYS. 2) The marker motions tracked from the real sensor's data have the noise in marker extraction and tracking. 3) When shear loads are present, our model cannot model the partial slip but it is very common for the real contact cases.

Speed Testing Our dense marker motion field simulation runs 9.22 seconds on average tested with CPU only. The FEM simulation in ANSYS with CPU costs 2 to 4 hrs for difference cases. In addition, according to Narang et al. [23]'s 5.57 seconds per sim for BioTac sensor in Isaac Gym with GPU acceleration, our simulation has a reasonable low computing demand.

We show some final results that combine the optical simulation and marker motion simulation in Fig. 13.

V. CONCLUSION

We present Taxim, an example-based GelSight tactile simulation model that combines optical and marker motion field simulation. We construct a polynomial table to simulate the optical response of GelSight from the contact geometry, and apply the linear displacement relationship and the superposition principle to simulate the markers' motion. Our simulation is computationally light weight, easy to set up and use, and simple to apply to different sensors. It also incorporates the sensor's illumination features and system noise through calibration with examples from real sensors. We have shown that our optical simulation outperforms the other state-of-the-art tactile simulations, and our marker motion field simulation achieves high accuracy by evaluating on a self-designed dataset. To the best of our knowledge, this is the first integrated work considering both the optical and marker motion simulation.

To extend this work, we plan to apply different sim-to-real robot perception and manipulation tasks using our simulation model. In addition, currently we simulate the quasi-static contact, and we plan to investigate the dynamic contact process and simulate the dynamic phenomena such as slip. The simulation pipeline can be computationally improved by applying GPU acceleration.

ACKNOWLEDGMENT

This work is supported by Facebook. The authors sincerely thank Arpit Agarwal, Tim Man, Yufan Zhang, Xiaofeng Guo, Uksang Yoo, Raj Kolamuri, Ruijia Xing, William Yan, Alankar Kotwal, Ioannis Gkioulekas, Mononito Goswami, Sudharshan Suresh, Haomin Shi for all the help on the discussion, experiment setup and manuscript revision.

REFERENCES

- W. Yuan, S. Dong, and E. H. Adelson, "Gelsight: High-resolution robot tactile sensors for estimating geometry and force," *Sensors*, vol. 17, no. 12, p. 2762, 2017.
- [2] "Gazebo," http://gazebosim.org/.
- [3] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," 2016.
- [4] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2012, pp. 5026–5033.
- [5] R. Tedrake, T. Team, *et al.*, "Drake: Model-based design and verification for robotics," 2019.
- [6] J. Allard, S. Cotin, F. Faure, P.-J. Bensoussan, F. Poyer, C. Duriez, H. Delingette, and L. Grisoni, "Sofa-an open source framework for medical simulation," in *MMVR 15-Medicine Meets Virtual Reality*, vol. 125. IOP Press, 2007, pp. 13–18.
- [7] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," *arXiv preprint arXiv:2108.10470*, 2021.
- [8] 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. IEEE, 2009, pp. 1070– 1077.
- [9] I. Gkioulekas, A. Levin, F. Durand, and T. Zickler, "Micron-scale light transport decomposition using interferometry," ACM Transactions on Graphics (ToG), vol. 34, no. 4, pp. 1–14, 2015.



Fig. 13: Marker motion field simulation with optical simulation results. We visualize the marker motions (scaled up by 20 for better visualization) on the dataset under different normal displacements and shear displacements.

- [10] Z. Ding, N. F. Lepora, and E. Johns, "Sim-to-real transfer for optical tactile sensing," in 2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2020, pp. 1639–1645.
- [11] B. Ward-Cherrier, N. Pestell, L. Cramphorn, B. Winstone, M. E. Giannaccini, J. Rossiter, and N. F. Lepora, "The tactip family: Soft optical tactile sensors with 3d-printed biomimetic morphologies," *Soft robotics*, vol. 5, no. 2, pp. 216–227, 2018.
- [12] A. Agarwal, T. Man, and W. Yuan, "Simulation of vision-based tactile sensors using physics based rendering," in 2021 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2021, pp. 1–7.
- [13] 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.
- [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," arXiv preprint arXiv:2012.08456, 2020.
- [15] 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). IEEE, 2019, pp. 5418–5424.
- [16] C. Hui, S. Hanqiu, and J. Xiaogang, "Interactive haptic deformation of dynamic soft objects," in *Proceedings of the 2006 ACM international* conference on Virtual reality continuum and its applications, 2006, pp. 255–261.
- [17] Y. Wang, W. Huang, B. Fang, F. Sun, and C. Li, "Elastic tactile simulation towards tactile-visual perception," in *Proceedings of the* 29th ACM International Conference on Multimedia, 2021, pp. 2690– 2698.
- [18] D. Casas and M. A. Otaduy, "Learning nonlinear soft-tissue dynamics for interactive avatars," *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, vol. 1, no. 1, pp. 1–15, 2018.
- [19] Z. Pezzementi, E. Jantho, L. Estrade, and G. D. Hager, "Characterization and simulation of tactile sensors," in 2010 IEEE Haptics Symposium. IEEE, 2010, pp. 199–205.
- [20] S. Moisio, B. León, P. Korkealaakso, and A. Morales, "Simulation of tactile sensors using soft contacts for robot grasping applications," in 2012 IEEE International Conference on Robotics and Automation. IEEE, 2012, pp. 5037–5043.
- [21] Y. S. Narang, K. Van Wyk, A. Mousavian, and D. Fox, "Interpreting and predicting tactile signals via a physics-based and data-driven framework," *arXiv preprint arXiv:2006.03777*, 2020.
- [22] T. Yamamoto, N. Wettels, J. A. Fishel, C.-H. Lin, and G. E. Loeb, "Biotac -biomimetic multi-modal tactile sensor," *Journal of the Robotics Society of Japan*, vol. 30, no. 5, pp. 496–498, 2012.
- [23] Y. Narang, B. Sundaralingam, M. Macklin, A. Mousavian, and D. Fox, "Sim-to-real for robotic tactile sensing via physics-based simulation and learned latent projections," *arXiv preprint arXiv:2103.16747*, 2021.
- [24] 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.

- [25] C. Sferrazza, T. Bi, and R. D'Andrea, "Learning the sense of touch in simulation: a sim-to-real strategy for vision-based tactile sensing," in 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2020, pp. 4389–4396.
- [26] F. R. Hogan, M. Jenkin, S. Rezaei-Shoshtari, Y. Girdhar, D. Meger, and G. Dudek, "Seeing through your skin: Recognizing objects with a novel visuotactile sensor," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 1218–1227.
- [27] 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.
- [28] W. Yuan, R. Li, M. A. Srinivasan, and E. H. Adelson, "Measurement of shear and slip with a gelsight tactile sensor," in 2015 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2015, pp. 304–311.
- [29] R. Calandra, A. Owens, M. Upadhyaya, W. Yuan, J. Lin, E. H. Adelson, and S. Levine, "The feeling of success: Does touch sensing help predict grasp outcomes?" in *Conference on Robot Learning*. PMLR, 2017, pp. 314–323.
- [30] A. Church, J. Lloyd, R. Hadsell, and N. F. Lepora, "Optical tactile sim-to-real policy transfer via real-to-sim tactile image translation," arXiv preprint arXiv:2106.08796, 2021.
- [31] S. Cotin, H. Delingette, and N. Ayache, "Real-time elastic deformations of soft tissues for surgery simulation," *IEEE transactions on Visualization and Computer Graphics*, vol. 5, no. 1, pp. 62–73, 1999.
- [32] S. Dong, W. Yuan, and E. H. Adelson, "Improved gelsight tactile sensor for measuring geometry and slip," in 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2017, pp. 137–144.
- [33] A. Hertzmann and S. M. Seitz, "Example-based photometric stereo: Shape reconstruction with general, varying brdfs," *IEEE Transactions* on Pattern Analysis and Machine Intelligence, vol. 27, no. 8, pp. 1254– 1264, 2005.
- [34] M. K. Johnson, F. Cole, A. Raj, and E. H. Adelson, "Microgeometry capture using an elastomeric sensor," ACM Transactions on Graphics (TOG), vol. 30, no. 4, pp. 1–8, 2011.
- [35] M. E. Angelopoulou and M. Petrou, "Uncalibrated flatfielding and illumination vector estimation for photometric stereo face reconstruction," *Machine vision and applications*, vol. 25, no. 5, pp. 1317–1332, 2014.
- [36] "Solidworks," https://www.solidworks.com/.
- [37] The GIMP Development Team, "Gimp." [Online]. Available: https://www.gimp.org
- [38] "Google Scanned Objects," https://app.ignitionrobotics.org/ GoogleResearch/fuel/collections/Google%20Scanned%20Objects.