3D Contact Point Cloud Reconstruction From Vision-Based Tactile Flow

Yipai Du, Guanlan Zhang, and Michael Yu Wang, *Fellow, IEEE*

***Abstract*—With the growing interest in vision-based tactile sensors, various types of sensors that utilize digital imaging are being developed. Among them, a group of sensors captures the tactile flow resulted from the contact deformation using the optical flow algorithm from computer vision and achieves full resolution deformation tracking on the tactile surface. In this work, a novel 3D contact reconstruction algorithm is proposed and evaluated. It exploits the contact geometry and projection relationship in the tactile flow, which are versatile for vision-based tactile sensors, unique for tactile perception but not inherited from computer vision. The resulted 3D contact point cloud representation is consistent with the tactile flow constraint, scale estimation, and contact edge estimation. It can be directly manipulated in downstream applications such as contact force estimation and contact pose estimation. Experiments and examples are provided that indicate the potential for the proposed tactile processing algorithm to connect tactile perception to tactile enabled robotic manipulation tasks.**

***Index Terms*—Contact modeling, force and tactile sensing, perception for grasping and manipulation.**

I. Introduction

THE sense of touch is critical for robotic systems to perceive the physical properties of objects for safe interaction by providing feedback for adaptation [1]. With the technological improvements in digital imaging and computer vision, vision-based tactile sensors using a camera to see through a transparent gel have become more popular. They provide superior sensing resolution and robustness to environmental changes. The most representative variants include GelSight [2], GelSlim [3], and Digit [4]. These sensors have made successful applications to enable more flexibility in dexterous manipulations [5], [6].

With an aim on the ease of manufacturing and availability, DelTact [7] adopts a modular design without strict illumination requirements and can be calibrated with minor efforts. The sensor takes the image stream of a deformable silicone gel (the tactile surface) with a dense random color pattern attached to it. One of the advantages is that the tactile flow, which is computed from the image using dense optical flow algorithm [8], is captured at high resolution and frequency. With the calibrated camera model, the search space of the 3D location for each image point is on a ray shot from the camera's optical center. However, it cannot be determined precisely due to the monocular scale ambiguity. Thus, the reconstruction of the 3D contact point cloud must be considered from a global point of view. Previous works have been mostly focused on learning-based methods for 3D contact estimation. The data required for training the neural networks can be from the real world by some automated process to save human labor or from simulated environments [9], [10]. In either case, data generation work needs to be done, and the transfer and generalization ability of the neural networks requires attention. With the image projection geometry analysis, we hope to achieve 3D contact estimation without a learned model. The image projection geometry model is more generalizable across devices and sensor types and requires little human effort.

In this letter, we propose to reconstruct the 3D contact point cloud with an optimization-based method. The problem is formulated as a convex optimization problem based on the image geometry model, which can be solved efficiently online. The solution to the optimization problem has the following advantageous properties:

- *Tactile flow constraint satisfaction:* The result is consistent with the measured tactile flow due to the problem construction.
- *Surface smoothness:* The total variation minimization smooths the reconstructed contact point cloud.
- *Depth consistency:* The optimization problem solves for the point cloud that agrees with the prior depth information.

Moreover, the optimization problem is model-based and involves no parameter-tuning, making it possible to generalize across sensors with less effort. A graphical overview of the obtained result is shown in Fig. 1 .

The remaining of the letter is structured as follows. In Section II, related works about vision-based tactile signal processing are introduced. In Section III, the 3D contact reconstruction problem is formulated as a convex optimization problem. In Section IV, the experiments both in finite element simulation and real-world are conducted. The finite element method (FEM) quantifies the reconstruction error using a random test indenter with the presence of different shear loads. The real-world indentation tests give a qualitative understanding of the versatility of

Manuscript received 23 May 2022; accepted 9 October 2022. Date of publication 14 October 2022; date of current version 20 October 2022. This letter was recommended for publication by Associate Editor Y. Jia and Editor A. Banerjee upon evaluation of the reviewers' comments. This work was supported in part by Hong Kong Innovation and Technology Fund (ITF) under Grant ITS/036/21FP and in part by the Project of Hetao Shenzhen-Hong Kong Science and Technology Innovation Cooperation Zone under Grant HZQB-KCZYB-2020083. *(Corresponding author: Yipai Du.)*

Yipai Du and Guanlan Zhang are with the Hong Kong University of Science and Technology, Hong Kong (e-mail: yipai.du@connect.ust.hk; gzhangaq@connect.ust.hk).

Michael Yu Wang is with the Hong Kong University of Science and Technology, Hong Kong, and also with the HKUST Shenzhen-Hong Kong Collaborative Innovation Research Institute, Shenzhen 518000, China (e-mail: michael.y.wang@monash.edu).

Digital Object Identifier 10.1109/LRA.2022.3214786

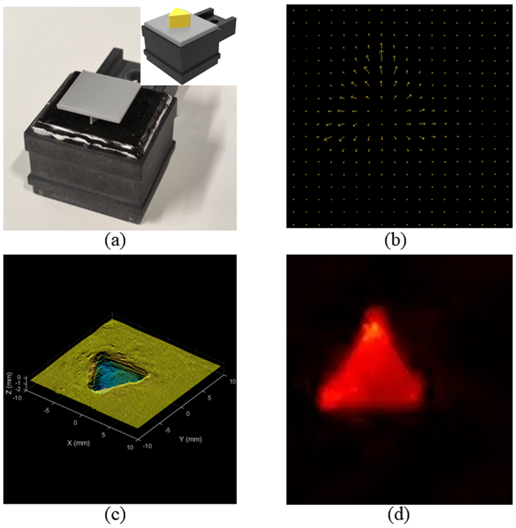


Fig. 1. (a) A triangular test indenter makes contact with the DelTact [7] tactile sensor. (b) The captured tactile flow (plotted sparsely for visualization). (c) The reconstructed 3D contact point cloud. (d) The depth map extracted from (c).

the proposed method. The contact force estimation and object pose estimation experiment give example applications of how the reconstructed 3D contact point cloud can be utilized. In Section V, the conclusion and future work are summarized.

II. Related Works

Tactile sensing is crucial for animals to adapt to the environment. Therefore, it has also been considered critical for robotics research to make robots behave reactively. Tactile sensors with resistive, capacitive, and piezoelectrical materials as transduction interfaces usually suffer from sensitivity to environments (e.g., temperature variation and electrical interference) and complicated wiring due to electric single point signals involved [11].

Vision-based tactile sensing approaches are becoming promising in the past two decades due to their better sensing resolution, easy manufacturing method, robustness in harsh settings, and multi-axial measuring capability. A soft surface that is responsive to contact is used, which adds robustness to contact uncertainty with its high-friction and compliant nature [12]. When it deforms, the shape change is captured by the camera in the 2D image. Since the birth of vision-based tactile sensing, a key question has been how to extract 3D information from this 2D image. The pioneers of vision-based tactile sensing, Kamiyama et al. [13], distributed dot markers with different colors at different depths for a clue in the normal direction by recording the positional center of mass variation in the markers. After that, other approaches to solving tactile sensing emerged. The GelSight [14], GelSlim [15] and Digit [4] sensors use photometric stereo to map from RGB color to surface normal for depth map reconstruction. Bauza et al. [16] and Suresh et al. [17] employed convolutional neural networks to learn the full depth from images, which gave more accurate results than the lookup table method. However, they potentially required a large dataset across different shapes. Wang et al. proposed to use MLP (multilayer perceptron) to learn the mapping from (r, g, b, x, y) to surface normals [18], and Sodhi et al. [19] adopted the pix2pix model. These two models are lightweight and, therefore, more data efficient. Dot markers are required for the photometric type of sensors to allow measurements in the shear direction [2]. Nevertheless, the dot markers would compromise the depth reconstruction accuracy and efficiency [18]. Our proposed method is based on Deltact sensor [7], which gives the full resolution 2D deformation flow field, i.e., the tactile flow, at high speed, with a commonly seen RGB camera. Thus, it avoids the interference of shear and geometry at the design phase. Moreover, the Deltact sensor features a model-based sensing principle, for which the deformation geometry is measured. Hence it is possible to exploit the geometry to reduce or even eliminate the large data requirement. *Soft-bubble* [20] grippers use a compact ToF (Time of Flight) depth sensor to enable contact shape reconstruction and object pose estimation. A pseudorandom dot pattern on the interior of the bubble surface enables extracting tangential shear force. Ambrus et al. [21] replaced the ToF sensor with a neural network to learn to predict depth from the IR image, which has a great potential to reduce the form factor and cost of the tactile gripper. However, the expensive ToF sensor is still required at the data collection stage, which is not easily accessible. The TacTip family [22] tracks the movements of an array of pins on the inside of the sensing surface and uses learning-based methods to solve various robotic estimation problems at high precision. Sferrazza et al. [23] adopted randomly scattered particles as tracking targets and combined dense optical flow with a neural network with a focus on accurate normal and shear force estimation [9]. Cui et al. developed GelStereo [24] that solved the depth estimation problem with learning-based stereo matching methods.

Although learning-based processing is prospering in tactile sensing, few people have investigated optimization-based methods. As computational power grows and algorithm efficiency improves, solving medium-scale convex problems online is possible and commonly seen in many signal processing applications [25]. One successful example in tactile sensing is by Kuppuswamy et al. who tried to solve another problem: estimating the actual external object shape in contact from a highly deformed and compliant surface using model-based Quadratic Program (QP) [12]. A research gap in the tactile sensing community is that all the works mentioned above attempted to solve the 3D contact tracking problem over-complicated. They either use more than one sensing modality or rely on machine learning to deal with the 2D to 3D mapping. Our discovery in this work is that the contact shape can be roughly estimated by incorporating the image projection geometry and the surface smoothness. In the presence of some scale prior (Gaussian density [26] in our case), the absolute scale can also be estimated, making the 3D tracking good enough for many robotic applications.

III. System Setup and Problem Formulation

The proposed method is based on the vision-based tactile sensor developed in our lab [7]. The sensor captures images of a specially designed colored pattern and processes them to obtain

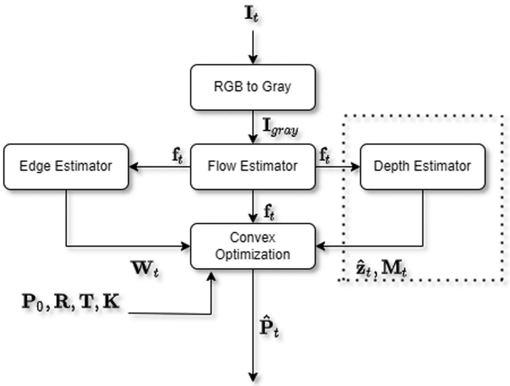


Fig. 2. The pipeline for solving the 3D point cloud reconstruction problem. Arrows denote signals and blocks denote processing modules. The dotted region is possible to be replaced with some other method or sensing modality to provide depth information to constrain the optimization.

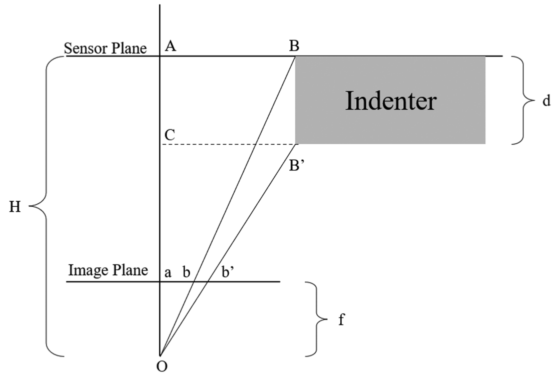


Fig. 3. A simplified projection graph for the sensor surface deformation resulted from a rectangle indenter. After the indentation, point B on the original sensor surface moves to point B'. This results in point b in the image going to point b'. The indentation depth is d, and the camera's focal length is f. The distance from the sensor plane to the optical center is denoted as H.

the optical flow resulted from contact deformation at a high frequency and fidelity. We name the measured optical flow from the images as the tactile flow and use the name throughout this letter because it is the flow field that encodes all the tactile information. The intrinsic parameter $\mathbf{K}$ and extrinsic parameters $\mathbf{R}$ and $\mathbf{T}$ are obtained through the sensor calibration process using printed chessboards with multiple poses, which is detailed in [7]. The sensor captured images are undistorted. Therefore no distortion parameter is needed. During the calibration process, the 2D-3D correspondences $(u, v) \leftrightarrow (x, y, z)$ are established. Given the fact that 3D points at rest lie on the same plane, the mapping from 3D coordinates $\mathbf{P}_0 = \{(x_i, y_i, z_i)\}$ to pixel coordinates (u_i, v_i) is obtained using a linear regression model. $\mathbf{P}_0$ is taken as the initial position of the cloud points. In practice, 100×100 points are used, and the point coordinates are flattened into a vector $\mathbf{P}_0 \in \mathbb{R}^{30000}$.

The goal of the contact point cloud reconstruction is, given the calibration parameters $\mathbf{K}$, $\mathbf{R}$, $\mathbf{T}$, the initial point locations $\mathbf{P}_0$ and the current tactile flow $\mathbf{f}_t$, estimating the current 3D point locations $\hat{\mathbf{P}}_t$. It is an ill-posed problem because the solution to the problem is not unique due to the monocular setting. However, the movement of the 3D point cloud is constrained in the sensor space and should exhibit smoothness. Utilizing this prior knowledge helps to overcome the monocular ambiguity.

The 3D contact reconstruction problem is divided into 3 sub-modules: contact edge estimation, depth estimation, and convex optimization, as shown in Fig. 2. They will be discussed in detail in the remaining of this letter. The contact edge estimation produces contact-edge-aware smoothness weights. The depth estimation reveals the overall scale (level of depth) of the contact. The optimization problem searches for an edge-aware smooth point cloud that simultaneously satisfies the tactile flow and depth prediction. Therefore, the solution is considered a reasonable estimate of the actual contact point cloud.

A. Contact-Edge-Aware Smoothness Weights

One feature in the resulted contact point cloud is the smoothness, but smoothing uniformly is naive and will produce an overly smooth result. The boundary between the contact and the non-contact region, i.e., the contact edge, needs to be preserved via an edge-aware smoothness weight. One simple choice is doing edge detection in the tactile flow. However, it is theoretically unsound. Fig. 3 illustrates a simplified case for the contact deformation with a rectangle indenter. Using similar triangles $\triangle$OAB and $\triangle$Oab, we have

$$\frac{\overline{ab}}{\overline{AB}} = \frac{\overline{Oa}}{\overline{OA}} = \frac{f}{H}. \tag{1}$$

Likewise, in $\triangle$OCB' and $\triangle$Oab,' there is

$$\frac{\overline{ab'}}{\overline{AB}} = \frac{\overline{ab'}}{\overline{CB'}} = \frac{\overline{Oa}}{\overline{OC}} = \frac{f}{H-d}. \tag{2}$$

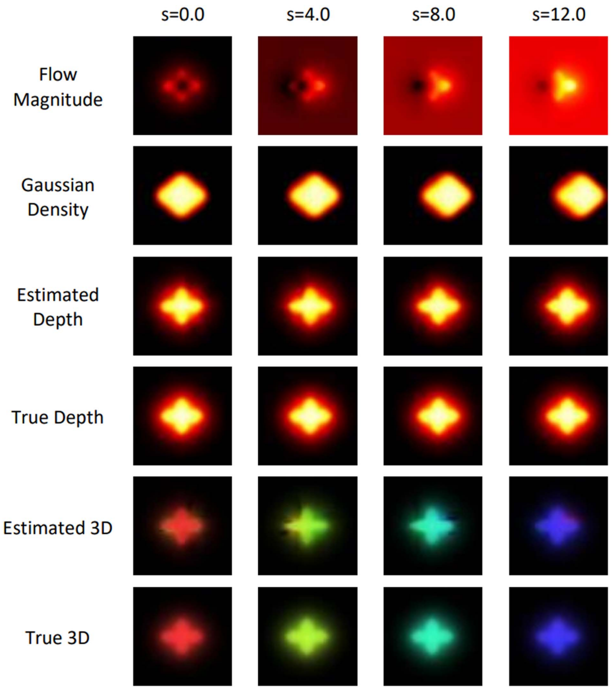
Combining (1) and (2), we have

$$\overline{ab'} - \overline{ab} = \frac{df}{H-d}\overline{AB}. \tag{3}$$

The left hand side of (3) is the resulted tactile flow in the image. From (3), it can be concluded that with the same deformation depth, the tactile flow increases linearly with respect to the distance to the center of the image. Therefore, the quantity

$$\mathbf{D} = \sqrt{\left(\frac{\partial \mathbf{f}_x}{\partial x}\right)^2 + \left(\frac{\partial \mathbf{f}_y}{\partial y}\right)^2} \tag{4}$$

can serve as an indicator for the depth (correlated with depth) in the 2D image case, where $\mathbf{f}_x$ and $\mathbf{f}_y$ are the x and y components of the tactile flow, respectively. By observing $\mathbf{D}$ one can find that it is equivariant to uniform shear, meaning a uniform shear load will not change the value of $\mathbf{D}$ but only its location. The Gaussian density [26] is a smoothed version of $\mathbf{D}$, which mathematically explains why the Gaussian density can work well in estimating the contact depth. However, the over-smoothness of Gaussian density causes defects at contact boundaries, as can be seen in Fig. 4. Hence we refer to $\mathbf{D}$ but not the Gaussian density for the



	s=0.0	s=4.0	s=8.0	s=12.0
X-axis Error	0.202	0.180	0.156	0.124
Y-axis Error	0.165	0.183	0.132	0.101
Z-axis Error	0.093	0.094	0.069	0.053

Fig. 4. The result with FEM simulated data. The estimated and true depths refer to the z axis displacement of the point cloud. For the flow magnitude, the Gaussian density, and depth, brighter color means a larger numerical value. For the 3D point cloud, the displacements in XYZ axes are regarded as the HSV color channels, respectively.

contact edge estimation and let

$$\mathbf{W} = \begin{bmatrix} \frac{1}{|\frac{\partial \mathbf{D}}{\partial x}|} \\ \frac{1}{|\frac{\partial \mathbf{D}}{\partial y}|} \\ \frac{1}{|\frac{\partial \mathbf{D}}{\partial x}|} \\ \frac{1}{|\frac{\partial \mathbf{D}}{\partial y}|} \\ \frac{1}{|\frac{\partial \mathbf{D}}{\partial x}|} \\ \frac{1}{|\frac{\partial \mathbf{D}}{\partial y}|} \end{bmatrix} \tag{5}$$

to be the weight factor to be multiplied with the gradient of the estimated 3D displacement ((6)). Here $|\frac{\partial \mathbf{D}}{\partial x}| \in \mathbb{R}^{9900}$ (100 × 99 due to reducing one column) and $|\frac{\partial \mathbf{D}}{\partial y}| \in \mathbb{R}^{9900}$ (99 × 100 due to reducing one row) therefore we have $\mathbf{W} \in \mathbb{R}^{59400}$. The physical meaning is that large gradient of deformation is allowed if the gradient of $\mathbf{D}$ is large while the gradient of deformation is penalized where the gradient of $\mathbf{D}$ is small.

B. Depth Estimation

Due to the monocular camera setting, there is scale ambiguity in the $3D$ perception. By measuring the density distribution, the absolute scale of contact depth can be estimated [26]. In this work, the approximate depth map $\hat{\mathbf{z}}_t$ is obtained from the Gaussian density using a regression model. Then $\mathbf{M}_t$ is the mask with 1 at the point with maximal contact depth and 0 everywhere else. It is used to constrain only on the maximal contact depth of the estimated contact point cloud $\hat{\mathbf{P}}_t$. For the ease of deduction and verification, only single contact case is presented in this letter. However, the method can be generalized to multiple contacts by firstly clustering on the predicted depth to identify locations and numbers of those contact. Note that the depth estimation and contact mask can be other feature extraction techniques or sensing modalities. This means to replace the dotted region in Fig. 2 with other methods. For example, an embedded range sensor or MEMS sensor [27] can provide the single point contact measurement at specific locations of the tactile surface. Then $\mathbf{M}_t$ will give nonzero weights at the measurement locations. The problem defined in (6) serves as a way to fuse the sparse contact signal with the tactile flow to obtain the full resolution contact point cloud.

C. The Optimization Problem

The optimization finds the point cloud $\hat{\mathbf{P}}_t$ that solves the following problem

$$\begin{aligned} \min_{\hat{\mathbf{P}}_t} \quad & \|\mathbf{L}(\hat{\mathbf{P}}_t - \mathbf{P}_0) \odot \mathbf{W}_t\| \\ \text{s.t.} \quad & \mathbf{A}_z(\hat{\mathbf{P}}_t - \mathbf{P}_0) \leq 0 \\ & \mathbf{A}_z(\hat{\mathbf{P}}_t - \mathbf{P}_0) \geq \min(\hat{\mathbf{z}}_t) \\ & h(\hat{\mathbf{P}}_t) - h(\mathbf{P}_0) = \mathbf{f}_t \\ & \mathbf{M}_t \odot \mathbf{A}_z(\hat{\mathbf{P}}_t - \mathbf{P}_0) = \mathbf{M}_t \odot \hat{\mathbf{z}}_t \end{aligned} \tag{6}$$

where $\mathbf{L} \in \mathbb{R}^{59400 \times 30000}$ is the Laplacian operator that maps the point cloud displacement $\hat{\mathbf{P}}_t - \mathbf{P}_0$ into its spatial gradient, meaning

$$\mathbf{L}(\hat{\mathbf{P}}_t - \mathbf{P}_0) = \begin{bmatrix} \frac{\partial}{\partial x}(\hat{\mathbf{P}}_{t,x} - \mathbf{P}_{0,x}) \\ \frac{\partial}{\partial y}(\hat{\mathbf{P}}_{t,x} - \mathbf{P}_{0,x}) \\ \frac{\partial}{\partial x}(\hat{\mathbf{P}}_{t,y} - \mathbf{P}_{0,y}) \\ \frac{\partial}{\partial y}(\hat{\mathbf{P}}_{t,y} - \mathbf{P}_{0,y}) \\ \frac{\partial}{\partial x}(\hat{\mathbf{P}}_{t,z} - \mathbf{P}_{0,z}) \\ \frac{\partial}{\partial y}(\hat{\mathbf{P}}_{t,z} - \mathbf{P}_{0,z}) \end{bmatrix} \tag{7}$$

where the subscript x y and z refer to the 3 components of the point cloud. $\odot$ denotes the Hadamard element-wise product to weight the spatial gradient with $\mathbf{W}_t$. Although $\mathbf{L}$ is large in dimension, it is very sparse, involving only $+1$, -1, and a large number of zeros, which means it can be handled by many efficient solvers that can manipulate sparse linear algebra. The minimization objective represents a contact edge-aware smoothness cost for the contact surface gradient. The matrix $\mathbf{A}_z$ extracts out all the z components in $\hat{\mathbf{P}}_t - \mathbf{P}_0$ therefore $\mathbf{A}_z(\hat{\mathbf{P}}_t - \mathbf{P}_0) \in \mathbb{R}^{10000}$. The first two inequalities constrain the z-axis deformation within the sensor space and do not exceed the largest estimated deformation depth (i.e., $\min(\hat{\mathbf{z}}_t)$). The first equality constraint involves a projection function $h : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ with $\mathbf{K}$, $\mathbf{R}$ and $\mathbf{T}$ as its parameters. It is not a linear function but a monomial, making the optimization become a geometric

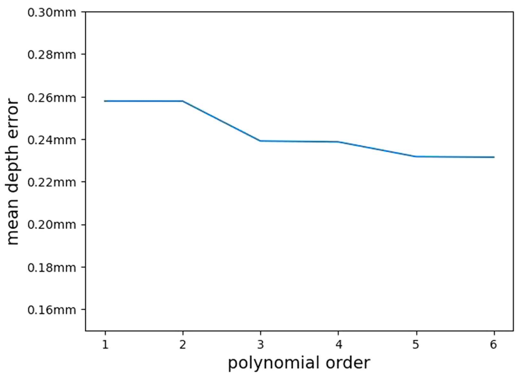


Fig. 5. The depth regression error from the Gaussian density using polynomial order up to 6.

program. To reduce the constraint complexity, now first consider a simplified case where there is only one point to be tracked, i.e. $\mathbf{P} \in \mathbb{R}^{3\times 1}$.

$$\begin{bmatrix} p_x \\ p_y \\ s \end{bmatrix} = \mathbf{K}(\mathbf{RP} + \mathbf{T})$$

$$h(\mathbf{P}) = \begin{bmatrix} \frac{p_x}{s} \\ \frac{p_y}{s} \end{bmatrix} = h(\mathbf{P}_0) + \mathbf{f}_t$$

$$\begin{bmatrix} p_x \\ p_y \end{bmatrix} = s(h(\mathbf{P}_0) + \mathbf{f}_t). \tag{8}$$

Since $h(\mathbf{P}_0)$ and $\mathbf{f}_t$ are just numerical parameters, only p_x, p_y and s are optimization variables, where p_x and p_y are unnormalized homogeneous coordinates in the image and s is their corresponding scale. This constraint is translated into a linear equivalence. The last equality constraint in (6) provides exact depth measurements at certain locations encoded by $\mathbf{M}_t$. Eq. (6) states a norm minimization problem with linear constraints, which is a convex optimization problem. $\ell 2$ norm is used in our experiment.

With the advancement of the computation power and efficient solvers, the convex optimization problem with variables at $10\,k$ scale can be solved real-time on a generic processor [25]. Our naive implementation in CVXPY [28] takes 750 ms to solve the optimization problem with only a single CPU core. More specialized and commercial solvers can significantly improve the speed to meet the real-time requirement.

IV. Experiments and Results

In this section, we evaluate the proposed contact reconstruction method's performance, quantitatively using simulations in FEM and qualitatively through various real-world indentation tests. 2 typical applications that use the reconstructed contact point cloud, including contact force and pose estimation, are demonstrated, suggesting the proposed method's potential in various robotic applications.

A. Finite Element Example - An Ideal Case

We first use the Finite Element Method (FEM) to simulate an ideal case. The FEM setup contains 100×100 nodes with 1 unit length spacing for neighboring nodes. Since the sensing area of the sensor is 21 mm×21 mm, each unit length would correspond to approximately 0.21 mm in the real world. The material is set to be hyperelastic with Poisson's ratio of 0.35. A relatively complex, snowflake-shaped flat rigid indenter (same as used in Fig. 7 of [26]) makes contact with the soft material and moves downward by 5 units in the FEM simulation space. The 3D nodal displacements are projected onto the image plane with extrinsic camera parameter $\mathbf{R} = \mathbf{I}$ and $\mathbf{T} = [0, 0, 10]^\intercal$. During the indentation process, the relationship between Gaussian density and the depth is fitted with a linear regression model to give a predictive depth prior. In addition to the pure normal indentation, a uniform shear strain s is added in the positive x-direction with 4.0 units, 8.0 units, and 12.0 units, respectively, to examine the effectiveness of the proposed method under coupled normal and shear load. Fig. 4 summarizes the results. Compared to the Gaussian density prediction from our previous result [26], the depth prediction from the proposed method forms a contact boundary almost the same as the ground truth. Nevertheless, if just observing the tactile flow or the Gaussian density, it is hard even for a human to conclude this precise contact shape. The mean 3D error is defined as

$$\text{mean 3D error} = \frac{1}{3\,N} \sum_{i=0}^{N} \sum_{a\in\{\text{x,y,z}\}} |\hat{\mathbf{P}}_{i,a} - \mathbf{P}_{i,a}| \tag{9}$$

where $\hat{\mathbf{P}}$ is the estimated point cloud location and $\mathbf{P}$ is the simulated ground truth. The estimated 3D point cloud is very close to the ground truth for all the cases, with the z-axis being predicted more accurately thanks to the Gaussian density prediction. The error in shear direction is larger because the node points exhibit lateral movements due to the local stretch from the indenter even with a pure normal load. This effect is not well-captured by the estimation process. However, the shear error gets smaller when the actual shear load increases, making the shear load dominate the estimation process. All the errors are small considering the 1 unit initial spacing for the points, 5 unit normal load, and 12 unit maximum shear load in the FEM simulation coordinate, where each unit is equivalent to 0.21 mm in the real world.

B. Gaussian Density Regression

A good depth predictor with Gaussian density gives a more accurate depth prior for a better contact point cloud reconstruction result. Therefore, instead of straight-line fitting as in [26], a polynomial fit is used with more data collected. From the analysis in Section III-A, the Gaussian density should be independent of the location. Therefore in the polynomial model, the feature contains only the Gaussian density. A dataset with different indenters is collected with an automated 3-axis linear stage. The data collection details are discussed in [7]. Specifically, 5 different sized spherical indenters with diameters of 10, 12, 15, 18, and 22 mm made contact with the tactile sensor at different depths. Each

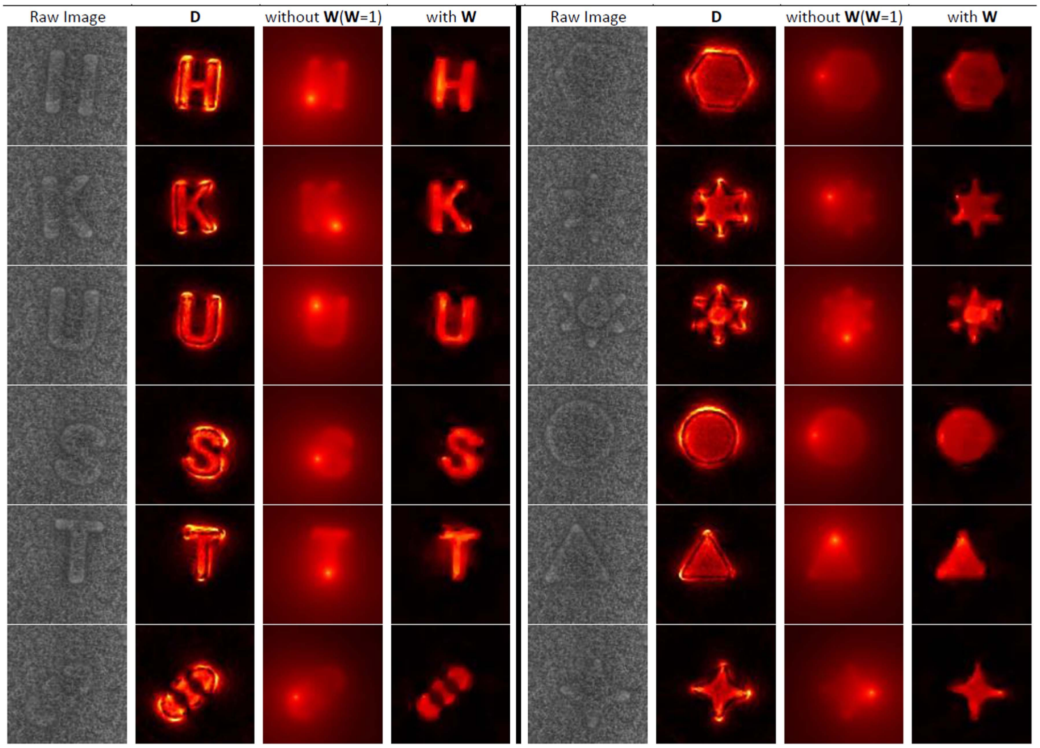


Fig. 6. The result from multiple different shapes making contact with the tactile sensor. The first column of each shape is the sensor-captured image. The second column shows the computed $\mathbf{D}$ as in Section III-A. The third column gives the reconstructed contact depth without the edge-aware weight $\mathbf{W}$ (i.e., setting $\mathbf{W} = 1$). The fourth column gives the estimated contact depth with the use of $\mathbf{W}$.

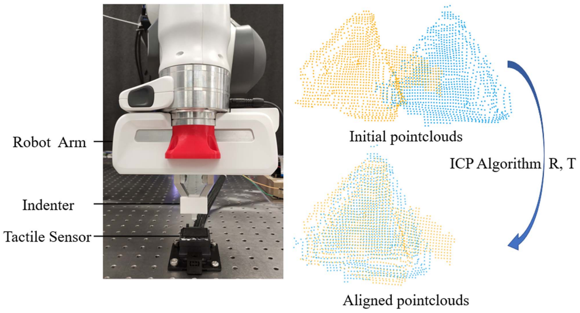


Fig. 7. The experimental setup of the pose estimation experiment.

contact was added with multiple shear loads to increase data variation. The total number of data points is 2070, 80% of which is used to train the polynomial regression and the remaining for testing. From Fig. 5, it can be seen that the error for the depth prediction decreases insignificantly when the polynomial order is greater than 3. Therefore we use order 3 in practice for better generalization, which has a mean error of depth 0.239 mm. Note that the true depth is in the set of $\{0.0, 0.5, 1.0, 1.5, 2.0\}$ mm. Therefore the depth error of 0.239 mm is within an acceptable range.

C. *Shape Test With Various Indenter*

Fig. 6 gives some estimated contact depth using the proposed method. It can be seen that although from the analysis in Section III-A, $\mathbf{D}$ should be constant for flat contact surfaces, in practice, it is often found to drop inside the contact contour, making the near boundary part looks brighter than the inner part. The reason is that, for $\mathbf{D}$ to be constant, the tactile flow $\mathbf{f}$ should increase/decrease linearly with respect to both x and y axes. It contradicts the local smoothness assumption [29] in calculating the optical flow: the flow vector should be close in a neighborhood. Thus in practice, the obtained tactile flow is often not so sharp to vary linearly. Therefore $\mathbf{D}$ is often found fluctuating. The problem is overcome by smoothing in [26]. This sim to real gap requires caution if one tries to transfer the simulated displacement vectors directly to real optical flow vectors, like in [9]. This discovery also calls attention that directly transferring the visual processing algorithms, which are proved successful in the rigid world, may yield an inconsistent result in vision-based tactile processing. In this work, however, the discrepancy does not make significance, as the estimated contact depth remains flat owing to the problem formulation, shown in the fourth column of each test case in Fig. 6. The third column gives the ablation study on the use of $\mathbf{W}$. Without $\mathbf{W}$, the result will be too smooth that blur into one piece.

D. *Application I: Force Estimation Through Contact*

With a meaningful and consistent decomposition of the flow field into 3D deformation $\hat{\mathbf{P}}_t - \mathbf{P}_0$, the force estimation can be made easier and more accurate. We consider the original tactile flow $\mathbf{f}_t$, the nHHD (Natural Helmholtz-Hodge Decomposition) decomposed tactile flow [30] $\mathbf{d}, \mathbf{r}, \mathbf{h}$, where $\mathbf{f}_t = \mathbf{d} + \mathbf{r} + \mathbf{h}$ and the 3D deformation $\hat{\mathbf{P}}_t - \mathbf{P}_0$ as the input features. Two different

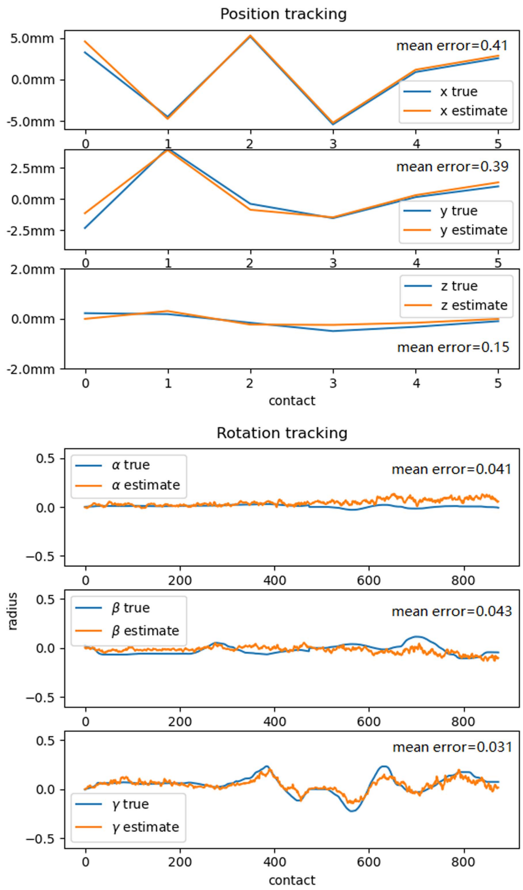


Fig. 8. The robot arm returned (regarded as ground truth) positional change and orientation change (represented in Euler angles) with the ICP estimates. For positional tracking, discrete contacts are made. For rotational tracking, contact is made solid and continuous. The mean tracking errors are shown on each plot.

machine learning models: linear regression Φ and MLP (multi-layer perceptron) Ω are employed, which are representatives of simple and complex machine learning models, respectively. For the linear regression, the goal is to find a linear function Φ for each of the following:

$$\begin{aligned}
\hat{F}_{raw,x} &= \Phi_{raw,x}\left(\sum \mathbf{f}_{t,x}\right),\\
\hat{F}_{raw,y} &= \Phi_{raw,y}\left(\sum \mathbf{f}_{t,y}\right)\\
\hat{F}_{raw,z} &= \Phi_{raw,z}\left(\sum \mathbf{f}_{t,x}, \sum \mathbf{f}_{t,y}\right)\\
\hat{F}_{nHHD,x} &= \Phi_{nHHD,x}\left(\sum \mathbf{f}_{t,x}\right)\\
\hat{F}_{nHHD,y} &= \Phi_{nHHD,y}\left(\sum \mathbf{f}_{t,y}\right)\\
\hat{F}_{nHHD,z} &= \Phi_{nHHD,z}\left(\sum \mathbf{d}_x, \sum \mathbf{d}_y\right)\\
\hat{F}_{3D,x} &= \Phi_{3D,x}\left(\sum(\hat{\mathbf{P}}_{t,x} - \mathbf{P}_{0,x}), \sum(\hat{\mathbf{P}}_{t,y} - \mathbf{P}_{0,y})\right)\\
\hat{F}_{3D,y} &= \Phi_{3D,y}\left(\sum(\hat{\mathbf{P}}_{t,x} - \mathbf{P}_{0,x}), \sum(\hat{\mathbf{P}}_{t,y} - \mathbf{P}_{0,y})\right)\\
\hat{F}_{3D,z} &= \Phi_{3D,z}\left(\sum(\hat{\mathbf{P}}_{t,z} - \mathbf{P}_{0,z})\right)
\end{aligned} \tag{10}$$

TABLE I
Force Estimation Evaluation

Model	RMSE (N)	F_x	F_y	F_z
Sum + Linear	Raw	1.28	1.25	5.72
	nHHD	1.28	1.25	5.55
	3D	**1.02**	**1.08**	**3.42**
MLP (1k×1k×1k)	Raw	0.326	0.408	1.41
	nHHD	0.325	0.278	0.882
	3D	**0.222**	**0.232**	**0.822**

Note that $\Phi_{3D,x}$ and $\Phi_{3D,y}$ use both x and y axis features because they are in the point cloud coordinate frame, which is not aligned with the camera/force coordinate frame. The summation works as a dimensionality reduction process and avoids overfitting for better generalization ability. For the MLP model Ω, the dimensionality is $1000 \times 1000 \times 1000$ with a dropout rate of 0.5, and the inputs are fed into Ω directly, i.e., $\mathbf{f}_t$ for raw, $\mathbf{d}, \mathbf{r}, \mathbf{h}$ for nHHD and $\hat{\mathbf{P}}_t - \mathbf{P}_0$ for 3D, respectively. The dataset used is the same as in Section IV-B, where the 3-axis force is collected simultaneously using an ATI Nano 17 F/T sensor. Table I lists the test RMSE for the force prediction. It can be seen that processing the tactile flow dataset into 3D contact point clouds outperforms the other two datasets in both machine learning models, with 3D+MLP performing best at predicting the contact force. This experiment proves that the estimated point cloud is not ad hoc but consistently decomposes the 2D tactile flow field into a 3D deformation field which is more simply correlated with the contact force. This application reveals that the proposed 3D contact reconstruction is suitable as a preprocessing technique for a simpler and more efficient contact force estimation.

E. Application II: Contact Pose Estimation From the Estimated Contact Point Cloud

One key advantage of tracking the contact point cloud is that it allows estimating the change of contact pose directly. As an example, a triangular indenter is fixed on the end effector of a Franka Emika Panda robot arm. The indenter contacts the table-fixed tactile sensor while the relative pose changes from the robot arm are recorded. Fig. 7 shows the experiment setup. A standard point-to-point ICP (iterative closest point) algorithm [31] is used to estimate the relative transformations between contacts. The experiment is divided into 2 parts. In the first part, the indenter contacts the tactile sensor at different locations on the sensor surface. In the second part, the indenter first contacts the tactile sensor and then rotates randomly to change its orientation while keeping the contact solid (non-slipping). The ICP algorithm tries to recover the positional change between contacts relative to the initial contact in the first case while estimating the orientation change in the second case. The results are plotted in Fig. 8. It can be seen that the positional estimation is very close to the actual value with only a sub-millimeter error while the object moves on the sensor surface. The orientation tracking is bearing larger relative errors, especially under significant rotation, since it is affected by the non-rigid structure of the indenter part and sensor gel distortion. Since there are errors from the robot arm pose estimates (sub-millimeter level) as well, this experiment is used just to roughly reflect the general trend, which shows the potential of the contact point cloud reconstruction to bridge to contact pose estimation. Possible future work can adopt a more accurate experimental setup and test cases aiming for more robust contact pose estimation.

V. Conclusion

This work proposes an optimization-based method for reconstructing the contact point cloud. The method is compatible with tactile flow type vision-based tactile sensors, such like [7] and [23]. The 3D contact reconstruction problem is formalized as an optimization problem incorporating the scale information (Gaussian density) and the tactile flow constraint. Quantitative experiments using the FEM simulated data demonstrate its ability to extract the contact structure with the fine resolution even when the contact is coupled with normal and shear loads. The qualitative testing with various indenters reveals that the proposed method can generalize to different contacts. Two extended applications of the obtained 3D contact are provided to illustrate the advantage of the obtained 3D contact representation, i.e., it simplifies the force estimation for better accuracy and allows direct contact pose estimation. The proposed method can be transferred to other sensors with similar principles, e.g., [23] and [20], as they also measure the dense tactile flow. Moreover, it is hoped that the processing framework gives insights into vision-based tactile processing. Unlike real-world vision tasks, which are more variational and dynamic, the tactile signals obtained from the camera embedded inside the vision-based tactile sensors are constrained, analyzable, and more monotonic. By studying the geometry in the deformation and projection process, the contact can be solved without complex machine learning algorithms and data collection. Machine learning can be auxiliary to this modeling process, like what we did in estimating the scale, but is only applied when necessary for better generalizable and transparent processing. We also identify the inconsistency issue in using optical flow algorithms, which are invented to track rigid movements, to track the tactile movement, as discussed in Section IV-C. In future research, distinctions of such kind between visual perception and tactile perception may require special attention to push the performance of vision-based tactile sensing to a higher level.

References

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

[2] 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, Art. no. 2762.

[3] 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 *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 1927–1934.

[4] 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. Automat. Lett.*, vol. 5, no. 3, pp. 3838–3845, Jul. 2020.

[5] Y. She, S. Wang, S. Dong, N. Sunil, A. Rodriguez, and E. Adelson, "Cable manipulation with a tactile-reactive gripper," *Int. J. Robot. Res.*, vol. 40, no. 12–14, pp. 1385–1401, 2021. [Online]. Available: https://doi.org/10.1177/02783649211027233

[6] S. Dong, D. K. Jha, D. Romeres, S. Kim, D. Nikovski, and A. Rodriguez, "Tactile-RL for insertion: Generalization to objects of unknown geometry," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 6437–6443.

[7] G. Zhang, Y. Du, H. Yu, and M. Y. Wang, "DelTact: A vision-based tactile sensor using a dense color pattern," *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 10778–10785, Oct. 2022.

[8] T. Kroeger, R. Timofte, D. Dai, and L. V. Gool, "Fast optical flow using dense inverse search," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 471–488.

[9] 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.

[10] Z. Si and W. Yuan, "Taxim: An example-based simulation model for GelSight tactile sensors," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 2361–2368, Apr. 2022.

[11] L. Zou, C. Ge, Z. J. Wang, E. Cretu, and X. Li, "Novel tactile sensor technology and smart tactile sensing systems: A review," *Sensors*, vol. 17, no. 11, 2017, Art. no. 2653.

[12] N. Kuppuswamy, A. Castro, C. Phillips-Grafflin, A. Alspach, and R. Tedrake, "Fast model-based contact patch and pose estimation for highly deformable dense-geometry tactile sensors," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 1811–1818, Apr. 2020.

[13] K. Kamiyama, H. Kajimoto, M. Inami, N. Kawakami, and S. Tachi, "A vision-based tactile sensor," in *Proc. IEEE Int. Conf. Artif. Reality Telexistence*, 2001, pp. 127–134.

[14] R. Li and E. H. Adelson, "Sensing and recognizing surface textures using a GelSight sensor," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 1241–1247.

[15] I. H. Taylor, S. Dong, and A. Rodriguez, "GelSlim 3.0: High-resolution measurement of shape, force and slip in a compact tactile-sensing finger," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2022, pp. 10781–10787.

[16] M. Bauza, O. Canal, and A. Rodriguez, "Tactile mapping and localization from high-resolution tactile imprints," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 3811–3817.

[17] S. Suresh, Z. Si, J. G. Mangelson, W. Yuan, and M. Kaess, "ShapeMap 3-D: Efficient shape mapping through dense touch and vision," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2022, pp. 7073–7080.

[18] S. Wang, Y. She, B. Romero, and E. Adelson, "GelSight wedge: Measuring high-resolution 3D contact geometry with a compact robot finger," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 6468–6475.

[19] P. Sodhi, M. Kaess, M. Mukadanr, and S. Anderson, "PatchGraph: In-hand tactile tracking with learned surface normals," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2022, pp. 2164–2170.

[20] N. Kuppuswamy, A. Alspach, A. Uttamchandani, S. Creasey, T. Ikeda, and R. Tedrake, "Soft-bubble grippers for robust and perceptive manipulation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 9917–9924.

[21] R. Ambrus, V. Guizilini, N. Kuppuswamy, A. Beaulieu, A. Gaidon, and A. Alspach, "Monocular depth estimation for soft visuotactile sensors," in *Proc. IEEE 4th Int. Conf. Soft Robot.*, 2021, pp. 643–649.

[22] N. F. Lepora, "Soft biomimetic optical tactile sensing with the TacTip: A review," *IEEE Sensors J.*, vol. 21, no. 19, pp. 21131–21143, Oct. 2021.

[23] C. Sferrazza and R. D'Andrea, "Design, motivation and evaluation of a full-resolution optical tactile sensor," *Sensors*, vol. 19, no. 4, 2019, Art. no. 928.

[24] S. Cui, R. Wang, J. Hu, C. Zhang, L. Chen, and S. Wang, "Self-supervised contact geometry learning by GelStereo visuotactile sensing," *IEEE Trans. Instrum. Meas.*, vol. 71, 2022, Art. no. 5004609.

[25] J. Mattingley and S. Boyd, "Real-time convex optimization in signal processing," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 50–61, May 2010.

[26] Y. Du, G. Zhang, Y. Zhang, and M. Y. Wang, "High-resolution 3-dimensional contact deformation tracking for FingerVision sensor with dense random color pattern," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 2147–2154, Apr. 2021.

[27] Y. Tenzer, L. P. Jentoft, and R. D. Howe, "Inexpensive and easily customized tactile array sensors using MEMS barometers chips," *IEEE Robot. Automat. Mag.*, vol. 21, no. 3, pp. 89–95, Sep. 2014.

[28] S. Diamond and S. Boyd, "CVXPY: A python-embedded modeling language for convex optimization," *J. Mach. Learn. Res.*, vol. 17, no. 83, pp. 1–5, 2016.

[29] A. Bruhn, J. Weickert, and C. Schnörr, "Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods," *Int. J. Comput. Vis.*, vol. 61, no. 3, pp. 211–231, 2005.

[30] Y. Zhang, Z. Kan, Y. Yang, Y. A. Tse, and M. Y. Wang, "Effective estimation of contact force and torque for vision-based tactile sensors with Helmholtz–Hodge decomposition," *IEEE Robot. Automat. Lett.*, vol. 4, no. 4, pp. 4094–4101, Oct. 2019.

[31] P. J. Besl and N. D. McKay, "Method for registration of 3-D shapes," *Proc. SPIE .*, vol. 1611, 1992, pp. 586–606.