

# Fast model-based contact patch and pose estimation for highly deformable dense-geometry tactile sensors

Naveen Kuppuswamy<sup>1</sup>, Alejandro Castro<sup>1</sup>, Calder Phillips-Grafflin<sup>1</sup>, Alex Alspach<sup>1</sup>, and Russ Tedrake<sup>1,2</sup>

**Abstract**—The difficulty of modelling deformable contact is a well-known problem in soft robotics and is particularly acute for compliant interfaces that permit large deformations, where the problem of inferring exact contact locations and manipuland pose is challenging. We present a model for the behavior of a highly-deformable *dense geometry* sensor in its interaction with objects; the *forward* model predicts the deformation of a mesh given the pose and geometry of a contacting rigid object. We use this model to develop a fast approximation to solve the *inverse* problem: estimating the contact patch when the sensor is deformed by arbitrary objects. This inverse model can be easily identified through experiments and is formulated as a sparse Quadratic Program (QP) that can be solved efficiently online. The proposed model serves as the first stage of a pose estimation pipeline for robot manipulation. We demonstrate the proposed inverse model through real-time estimation of contact patches on a contact-rich manipulation problem in which oversized fingers screw a nut onto a bolt, and as part of a complete pipeline for ICP-based pose-estimation and tracking. Our results demonstrate a path towards realizing soft robots with highly compliant surfaces that perform complex real-world manipulation tasks.

## I. INTRODUCTION

We imagine a future with robots that are able to make contact anywhere on their bodies in order to successfully execute tasks. Soft robot skins fundamentally change the mechanics of contact, act as low-pass filters for impacts, and potentially reduce the bandwidth requirements for control. Deformable skin changes point contacts into patch contacts which permit torsional friction, distribute loads across a larger surface area of the robot and object, and generally help to sustain contact. Soft skin can also provide passive robustness to geometric uncertainty, and as we demonstrate here, is conducive to rich tactile sensing.

Tactile sensing is increasingly used in robot manipulation since it affords a direct way of detecting the state of the manipuland through contact [1]. This is particularly relevant for manipulation in highly cluttered environments, such as homes, where vision-based sensors suffer from occlusions, varying lighting, and challenging material properties. Recent developments in the field have ushered in a new class of *dense-geometry* visuo-tactile sensors [2]–[4]. Novel materials coupled with imaging sensors capture high resolution geometric information that enables the estimation and measurement of contact geometry and locations, forces, slip, and other properties essential for precise manipulation. Observing contact geometry allows the tactile sensing problem to be formulated as a 3D-perception problem, for which a wide range of proven techniques exist. However, key issues with these sensors remain, including lack of sufficient mechanical

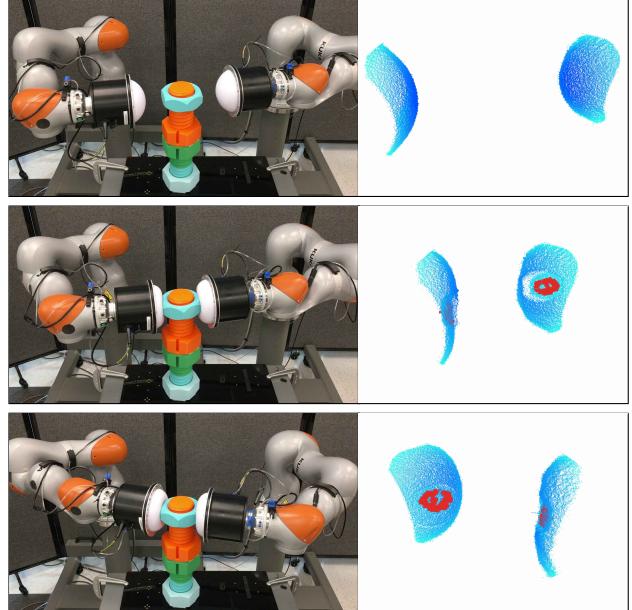


Fig. 1: Real-time contact patch tracking on the *Soft-bubble* sensor in the “dual-finger” manipulation task of screwing a nut on a bolt. The rectified point clouds from the sensors are shown in blue and the estimated contact patch in red.

robustness - they are subject to a large degree of wear and tear - and modelling difficulties in capturing their contact and deformation behavior [5]. Although data-driven techniques such as deep neural networks can help circumvent modeling difficulties, these are data (and sensor duty-cycle) intensive during the training phase. Development of first principles models of sensors not only enables simulation and analysis, but can be incorporated within model-based state-estimation and control frameworks.

Inspired by the human fingertip, the *Soft-bubble* [6] is a highly compliant, air-filled dense-geometry tactile sensor. These sensors have several ideal mechanical characteristics - they are durable, lightweight, and easy to fabricate and the compliant, high-friction surface produces stable contact patches between manipulator and manipuland. As a visuo-tactile sensor, it provides a high resolution depth image of the deformed contact surface. In previous work, these soft-bubbles have been used to classify manipulated objects as well as estimate their pose [6]. The extreme compliance of the *Soft-bubble* poses its own challenges, however. When an object deforms the membrane, deformation does not occur only at the patch in contact, but everywhere. It is difficult

to distinguish which parts of the deformation are directly caused by the manipuland and which parts are the result of internal air pressure and elastic membrane behavior. If these extraneous non-contact points are not filtered out, direct methods for pose-estimation using the depth image, such as Iterative Closest Point (ICP), often fail to find an accurate alignment.

In this paper, we present two key contributions towards addressing these challenges: (i) we present a *forward* model based on first principles of continuum mechanics to describe a highly-deformable air-filled membrane that makes contact with a rigid object of a given geometry; (ii) we then utilize this model to solve the *inverse* problem of identifying the contact patch based solely on the depth information from the sensor - we develop an approximate formulation to solve this problem using a sparse convex Quadratic Program (QP) which renders it solvable in real-time. While the forward model enables simulation and analysis, the inverse model enables online contact patch tracking necessary for a perception pipeline that performs object pose estimation.

We demonstrate the efficacy of the proposed method in two ways: (i) a dual-arm rich-contact manipulation problem in which an oversized nut is screwed onto a bolt as shown in Figure 1 where contact patches are tracked online, and (ii) combining the contact patch estimation with a conventional dual-stage ICP-based tracker to robustly estimate manipuland pose.

## II. BACKGROUND AND MOTIVATION

The field of tactile sensing has seen dramatic growth in the recent years [1] spurred by developments in novel fabrication techniques and signal processing methods [7]. Thus, its applications to dexterous robot manipulation have greatly expanded [8]. Increasingly, there has been an interest in integrating tactile sensing within state-estimation [9], often incorporating models of contact mechanics. A novel development in the field has been the growth of a class of visuo-tactile sensors that are ideal for characterizing the geometry of the object being sensed as well as identifying its pose relative to the manipulator. These high-resolution, *dense-geometry*, tactile sensors, such as GelSight [2], GelSlim [3], and FingerVision [4], use cameras to gather large amounts of data over relatively small contact areas and use either a combination of precise lighting and stereo algorithms, or machine learning to generate precise height maps of contacting geometry. This depth information is then utilized for pose estimation [10], force distribution estimation [5], [11] and feedback control for tactile tasks [12].

Difficulties in modelling contact mechanics, behavior of soft deformable surfaces, and handling large amounts of multi-modal information have often been cited as motivations for employing data-driven techniques in tactile state-estimation and control. Recent work seeks to use deep learning for predicting contact forces/slippage [13], [14], grasp stability identification [15], [16] or for direct end-to-end control [17].

Nevertheless, finding a first principles physical model of tactile behavior is appealing, since it often provides improved understanding, and allows integration into simulation, planning and control model-based frameworks. Moreover, current state-of-art machine learning methods often require large amounts of data in order to generalize sufficiently – generating a sufficient volume of training and test data can be impractical or even infeasible for sensors that need to make contact in order to sense. Many tactile sensors, including gel-based tactile sensors, are prone to degradation over such usage cycles.

Differing from gel-based visuo-tactile sensors, the *Soft-bubble* [6] utilizes an inflated opaque membrane with an off-the-shelf depth camera pointed at its interior surface. Due to its simple construction and its use of a naturally resilient elastic membrane, this sensor can withstand repeated large deformations without significant degradation. Further, its ability to deform around a contacting object more deeply than thin gel-based sensors results in a larger portion of the object’s geometry being captured in the output depth map. This sensor’s high resolution depth sensing, along with its ability to provide a large, high-friction contact patch make it an ideal contacting surface for manipulators.

We develop an efficient model-based formulation describing the deformation of a membrane subject to external contact. Most relevant to our approach, is a recent method for GelSlim sensors [5] that directly use measured deformations in a FEM model to estimate contact forces, an approach sensitive to experimental noise given it requires to compute second order derivatives from experimental data. Our approach solves this issue by computing a statistical best fit to experimental data constrained to satisfy the laws of physics.

We show in subsequent sections how this formulation can be employed within both a *forward* simulation as well as an *inverse* or sensing model that can capture the contact patch induced through contact between a soft sensor and the external world.

## III. SENSOR MODEL: MEMBRANE AS AN ASSEMBLY OF FLAT ELEMENTS

The soft contact surface of our sensor can be mathematically described as a membrane [18], [19]. The temporal dynamics of the membrane’s deformation are the result of fast time-scale traveling waves across the surface of the sensor; however, since these fast dynamics cannot be resolved by a 60 Hz depth camera, we can ignore membrane dynamics and instead use a quasi-static approximation describing the static equilibrium between internal stresses in the membrane and external contact forces. Thus our formulation does not involve time derivatives and does not introduce state. Since most of the deformation is caused by external forces pushing into the membrane, we neglect the effect of friction forces for the estimation of deformation. That is, the contact surface is modeled as *frictionless*. This is justified by the agreement with the experiments presented in Section V.

We make a linear approximation to the deformation of the membrane from a *reference surface configuration*  $\mathcal{S}_0$  initially

pressurized at an absolute pressure  $P_0$ . The kinematics of the membrane is approximated such that each material point  $\mathbf{x}_0$  in the reference configuration  $S_0$  deforms into a point  $\mathbf{x}$  in the *current configuration*  $S$  according to

$$\mathbf{x} = \mathbf{x}_0 + u(\mathbf{x}_0)\hat{\mathbf{n}}_0(\mathbf{x}_0), \quad (1)$$

where  $\hat{\mathbf{n}}_0(\mathbf{x}_0)$  is the outward normal to the surface *in the reference configuration* and  $u(\mathbf{x}_0)$  is a scalar field describing displacements in the normal direction only. Full mathematical descriptions [18], [19] also incorporate in-plane displacements and can describe a much richer range of deformations such as wrinkling and folding. The approximation in terms of a single scalar field  $u(\mathbf{x}_0)$  is justified by the observation that, when pressurized, the membrane of the sensor does not wrinkle or fold onto itself unless extreme contact loads are applied.

To further simplify the model, a homogeneous and isotropic in-plane stress is assumed, similar to the surface tension at the interface between two fluids [20]. The *surface tension* at the reference configuration is then characterized by a single scalar  $\gamma_0$ , which must be determined as part of a system identification process as a function of  $P_0$ . For this particular state of stresses, the out-of-plane balance of momentum reduces to

$$2B\gamma_0 + P = 0 \quad \text{in } S, \quad (2)$$

with  $B$  being the mean curvature and  $P$  the applied absolute pressure, which is the well known Young-Laplace equation [18].

We discretize the reference surface  $S_0$  with a mesh of triangle elements  $\Delta_0^e$  defined by the nodal points  $\mathbf{x}_{0,i}$  of the mesh. The deformation of the surface is then described in a Lagrangian manner by the motion of the mesh nodal points  $\mathbf{x}_{0,i}$  in the reference surface  $S_0$  to nodal positions  $\mathbf{x}_i$  in the current surface configuration  $S$ . In the notation above,  $i$  indexes the nodes of the mesh. Similarly,  $e$  indexes all triangle elements of the mesh.

For small displacements from the reference configuration, the mean curvature of a flat membrane element can be approximated as  $2B = \nabla^2 u$ , with the Laplace operator  $\nabla^2(\cdot)$  taken in a local two-dimensional frame for the flat triangle element. Thus the momentum balance in Eq. (2) reduces to

$$\gamma_0 \nabla^2 u + p = 0 \quad \text{in } \Delta_0^e. \quad (3)$$

In Eq. (3),  $u(\mathbf{x})$  represents small displacements from the reference configuration and pressure  $p(\mathbf{x})$  represents deviations from the reference pressure  $P_0$ . That is, the absolute pressure is given by  $P = P_0 + p$ .

Discretization of Eq. (3) in a local element frame  $T^e$  by a standard Finite Element Method (FEM) leads to the local stiffness matrix  $\mathbf{k}_{a,b}^e$ , where indexes  $a, b$  refer to local nodal points in element  $e$ . The global stiffness matrix of the system  $\mathbf{K}$  is then obtained from each element stiffness matrix  $\mathbf{k}_{a,b}^e$  using the standard assembly procedure from FEM [21, §6.2.13]. The pressure term in Eq. (3) is easily

lumped to obtain a diagonal matrix multiplying the vector of pressure unknowns. The final discrete equation reads

$$\mathbf{K}\mathbf{u} = \text{diag}(\mathbf{A})\mathbf{p}, \quad (4)$$

where  $\mathbf{u}$  is the vector of displacements,  $\mathbf{p}$  is the vector of normal stresses,  $\mathbf{A}$  is a vector that at the  $i$ -th entry contains an area associated with the  $i$ -th node in the mesh and  $\text{diag}(\mathbf{v})$  denotes the operator that forms a diagonal matrix with the elements of  $\mathbf{v}$ . With  $n$  the number of nodes in the mesh,  $\mathbf{u}$ ,  $\mathbf{p}$  and  $\mathbf{A}$  are vectors in  $\mathbb{R}^n$ . The stiffness matrix  $\mathbf{K} \in \mathbb{R}^{n \times n}$  is symmetric and positive definite.

#### A. Internal Pressure

The normal loads in Eq. (4) are split into a single scalar pressure  $p_v$  internal to the sensor and a contact pressure at each point  $p_c$  external to the sensor as

$$\mathbf{K}\mathbf{u} = -\text{diag}(\mathbf{A})\mathbf{p}_c + \mathbf{A}\mathbf{p}_v, \quad (5)$$

where  $p_v$  and  $p_c$  denote deviations from a nominal value and are zero in the reference configuration. While the total absolute pressure inside the sensor  $P = P_0 + p_v$  must always be positive, pressure deviations  $p_v$  can either be positive, indicating an increase of pressure, or negative indicating a decrease in pressure. Contact pressure  $p_c$  however is always positive since contact with an external object can only push into the membrane.

The total absolute pressure inside the sensor relates to the volume of air  $V$  within the sensor by the ideal gas law as

$$P(V) = P_0 \frac{V_0}{V}, \quad (6)$$

where  $V_0$  is the volume of the sensor in the reference configuration. An expression for the volume in terms of the displacements  $\mathbf{u}$  follows from Gauss's theorem as the surface integral

$$V = \frac{1}{3} \int_S \mathbf{x} \cdot \hat{\mathbf{n}}(\mathbf{x}) d^2x, \quad (7)$$

To obtain a linear expression of  $V$  with the displacements Eq. (7) is approximated as

$$V \approx \frac{1}{3} \int_S \mathbf{x} \cdot \hat{\mathbf{n}}_0(\mathbf{x}_0) d^2x_0, \quad (8)$$

Using the parametric representation for  $\mathbf{x}$  in Eq. (1) leads to the desired expression for the volume as

$$V(\mathbf{u}) = V_0 + \frac{1}{3} \mathbf{A}^T \mathbf{u}, \quad (9)$$

which can then be used in Eq. (6) to update the pressure as a function of the deformations  $\mathbf{u}$ .

#### B. Contact Constraints

Contact constraints are necessary for simulation purposes. Consider an external rigid object located in a position that interferes with the location of the membrane in its reference configuration, as shown in Fig. 2. A point  $\mathbf{x}_{0,k}$  that in the reference configuration is located inside the object must necessarily move in order to avoid an impossible interpenetration. Since Eq. (1) establishes that points can only move

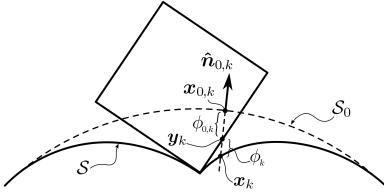


Fig. 2: Kinematics of contact between the deformable membrane of the sensor and an external rigid object.

in the normal direction, a potential contact point  $y_k$  on the object corresponding to  $x_{0,k}$  is determined by casting a ray from  $x_{0,k}$  in the direction opposite to the membrane's normal. The signed distance between point  $y_k$  and point  $x_k$  in the current configuration can be computed as

$$\phi_k = (\mathbf{y}_k - \mathbf{x}_k) \cdot \hat{\mathbf{n}}_{0,k} \quad (10)$$

which after substituting  $\mathbf{x}_k$  in Eq. (1) can be written as

$$\phi_k = (\mathbf{y}_k - \mathbf{x}_{0,k} - u_k \hat{\mathbf{n}}_{0,k}) \cdot \hat{\mathbf{n}}_{0,k} = \phi_{0,k} - u_k, \quad (11)$$

where  $\phi_{0,k}$  denotes the signed distance between  $\mathbf{y}_k$  and  $\mathbf{x}_{0,k}$ , a negative quantity indicating interpenetration. The set of all signed distances is placed within a single vector  $\phi$  of size  $n_c$ , the number of contact candidates. Equation (11) is written in vector form as

$$\phi = \phi_0 + \mathbf{H}\mathbf{u}, \quad (12)$$

with  $\phi$  and  $\phi_0$  in  $\mathbb{R}^{n_c}$  and  $\mathbf{H}$ , a sparse matrix in  $\mathbb{R}^{n_c \times n_c}$  with  $H_{k,i} = -1$  for the  $k$ -th contact corresponding to the ray cast from the  $i$ -th point  $x_{0,i}$  in the reference mesh. All other entries in  $\mathbf{H}$  are zero.

The contact constraints are a statement of the Signorini boundary condition. At each point on the membrane either the signed distance is zero and there is a positive contact pressure acting at that point or the signed distance is positive and the contact pressure at that point is zero since there is no contact. This condition is formally written as the complementarity condition

$$\mathbf{0} < \phi \perp \mathbf{p}_c > \mathbf{0}, \quad (13)$$

or, in terms of displacements

$$\mathbf{0} < \phi_0 + \mathbf{H}\mathbf{u} \perp \mathbf{p}_c > \mathbf{0}. \quad (14)$$

### C. Simulation

Given an external object to the sensor, both  $\phi_0$  and  $\mathbf{H}$  as defined in Section III-B are available as a function of the pose of the object relative to the sensor. In addition, given a mesh discretizing the reference configuration  $S_0$ , the stiffness matrix  $\mathbf{K}$  and the vector of nodal areas  $\mathbf{A}$  in Eq. (4) are fixed and can be pre-computed.

We observe that contact pressure in Eq. (5) can only be non-zero (positive) at candidate nodes  $k$  that in the reference configuration fall inside the external rigid object, see Fig. 2. That is, we can define a vector  $\mu \geq \mathbf{0}$  of size  $n_c$  such that  $-\text{diag}(\mathbf{A})\mathbf{p}_c = \mathbf{H}^T\mu$ . In other words,  $p_{c,i} = \mu_k/A_i$  for candidate nodes for which  $H_{k,i} = -1$  and  $p_{c,i} = 0$  otherwise.

Given the non-linear relationship between deformations and the internal sensor pressure given by Eqs. (6) and (9), we propose a fixed-point iteration on the internal pressure  $p_v$ . Given the pressure  $p_v^m$  at the  $m$ -th iteration, the following MLCP is solved for the unknowns  $\mathbf{u}^m$  and  $\mu_c^m$

$$\begin{aligned} \mathbf{K}\mathbf{u}^m &= \mathbf{A}p_v^m + \mathbf{H}^T\mu^m, \\ \text{s.t. } \mathbf{0} < \phi_0 + \mathbf{H}\mathbf{u}^m \perp \mu^m > \mathbf{0}. \end{aligned} \quad (15)$$

Note that we replaced  $-\text{diag}(\mathbf{A})\mathbf{p}_c = \mathbf{H}^T\mu$  in the momentum balance Eq. (5).

This problem is cast as the equivalent QP

$$\begin{aligned} \mathbf{u}^{m+1} &= \arg \min_{\mathbf{u}} \quad \frac{1}{2} \mathbf{u}^T \mathbf{K} \mathbf{u} - p_v^m \mathbf{A}^T \mathbf{u}, \\ \text{s.t. } \mathbf{0} < \phi_0 + \mathbf{H}\mathbf{u}. \end{aligned} \quad (16)$$

which states the Karush-Kuhn-Tucker (KKT) conditions for the QP in Eq. (15) where vector  $\mu^m$  takes the role of the KKT multipliers needed to enforce the inequality constraint in the QP.

Since  $\mathbf{K}$  is positive-definite the QP is convex and given  $\mathbf{H}$  has full rank the reduced Hessian of Eq. (16) is positive-definite and the QP has a unique solution. Note that the matrices  $\mathbf{K}$  and  $\mathbf{H}$  are sparse and therefore a QP solver that exploits sparsity is used.

Once  $\mathbf{u}^{m+1}$  is solved from Eq. (16), the enclosed air volume is updated with Eq. (9) and the internal pressure is updated according to the ideal gas law in Eq. (6). This provides the value of  $p_v^{m+1} = P^{m+1} - P_0$  for the next iteration and Eq. (16) is solved again with the updated value of pressure. Relaxation is used on the pressure  $p_v$  to attain convergence. A relaxation factor  $\omega = 0.6$  was found to work best by trial and error. Convergence is monitored on the relative change of  $p_v$  between iterations.

Figure 3 shows the simulation results for a rigid robot toy making contact with the sensor. Figure 3a renders the deformed membrane in yellow and the robot toy in a translucent green. The simulation is able to resolve the two separate indentations caused by each leg of the toy on the membrane. Note the bulging of the membrane between the legs of the toy caused by an increase of air pressure inside the sensor. Figure 3b shows a simulation of a point cloud for this configuration colored with the distance to the time-of-flight sensor with the soft membrane shown in translucent yellow for reference. The point cloud is simulated by ray casting from the center of the camera, in the direction of the ray  $\hat{\mathbf{r}}_r$  for each pixel, and computing the distance from the center of the camera to the point on the membrane hit by the ray. Gaussian noise is added to each distance in accordance with the noise levels reported by the manufacturer of the time-of-flight sensor. See Section IV-A for details on the simulation of the camera sensor.

### IV. INVERSE PROBLEM AND CONTACT PATCH DETERMINATION

The inverse problem consists on finding the patch of contact given we have a point cloud measured by an internal depth camera. This is nontrivial since the actual surface

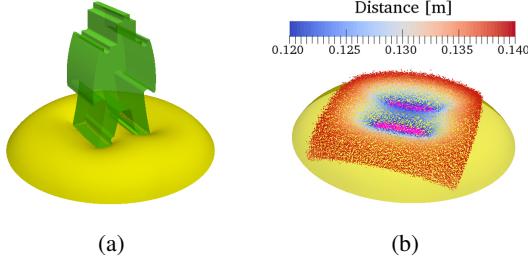


Fig. 3: Simulated soft membrane with robot toy pressed into it. (a) Rendering of the deformed sensor for a given toy pose; (b) Simulated point cloud colored based on distance to the time of flight sensor.

deformation extends to regions far out from the contact patch. This problem is aggravated as the external object pushes deeper into the sensor causing larger deformations. Therefore ad-hoc strategies that do not incorporate the physics of deformation are likely to perform poorly. For instance, we attempted to use a threshold on the point cloud distances. We found this strategy to be highly sensitive to threshold values even for simple convex geometries as in Fig. 7 and to perform very poorly in cases with complex non-convex geometries as shown in Fig. 6. More complex approaches to fit a mesh [22] and estimate curvatures require parameter tuning and amplify noise in the experimental data.

We propose an inverse problem strategy to find the external contact forces producing a deformation of the sensor that best matches the point cloud data. The overall formulation is a convex QP that minimizes the error between the distances measured by the real camera with those estimated by the model, while physics of deformation are enforced as linear constraints. In practice this strategy is shown to produce results with low levels of noise when compared to pure ad-hoc least square strategies, providing not only cleaner geometric measures but also a direct prediction of regions on which forces are applied.

#### A. Modeling Point Cloud Distances

The  $r$ -th point measured by the time of flight sensor is modeled as the point  $\mathbf{x}_r$  on the surface of the sensor in the current configuration that results from shooting a ray with direction  $\hat{\mathbf{r}}_r$  from the center of the camera  $\mathbf{c}$  towards the membrane. The ray direction  $\hat{\mathbf{r}}_r$  is a known fixed quantity for each pixel of the camera and can be found from the specifications of the given camera model.

From Eq. (1) we have

$$\mathbf{x}_r = \mathbf{x}_{0,r} + u_r \hat{\mathbf{n}}_{0,r}, \quad (17)$$

where  $u_r$  is the local displacement and  $\hat{\mathbf{n}}_{0,r}$  is the normal vector in the reference configuration.  $\mathbf{x}_{0,r}$  is found by ray-casting from the camera into the reference configuration. Therefore  $\mathbf{x}_{0,r}$  is computed once as a pre-processing step at the beginning of the computation together with the normal  $\hat{\mathbf{n}}_{0,r}$  and the FEM shape function vector  $\mathbf{S}_r$  such that the displacement at  $\mathbf{x}_{0,r}$  can be interpolated with  $u_r = \mathbf{S}_r^T \mathbf{u}$ .

The distance from the camera center  $\mathbf{c}$  to point  $\mathbf{x}_r$  is given by

$$d_r = \hat{\mathbf{r}}_r \cdot (\mathbf{x}_r - \mathbf{c}). \quad (18)$$

Using Eq. (17) and the pre-computed local interpolation  $u_r = \mathbf{S}_r^T \mathbf{u}$  for the displacement, the distance is written as

$$d_r = d_{0,r} + (\hat{\mathbf{r}}_r \cdot \hat{\mathbf{n}}_{0,r}) \mathbf{S}_r^T \mathbf{u}, \quad (19)$$

where  $d_{0,r} = \hat{\mathbf{r}}_r \cdot (\mathbf{x}_{0,r} - \mathbf{c})$  is the distance in the reference configuration  $\mathcal{S}_0$ . This equation is cast in vector form as

$$\mathbf{d} = \mathbf{d}_0 + \mathbf{D}\mathbf{u}, \quad (20)$$

with  $\mathbf{d}_0$  the vector of distances in the reference configuration,  $\mathbf{d}$  the vector of distances in the current configuration and matrix  $\mathbf{D}$  containing at each  $r$ -th row the row vector  $(\hat{\mathbf{r}}_r \cdot \hat{\mathbf{n}}_{0,r}) \mathbf{S}_r^T$ .

#### B. Inverse Problem Cost function

The inverse problem is formulated so that the error  $f = \|\mathbf{d} - \tilde{\mathbf{d}}\|^2$  between the measured distances  $\tilde{\mathbf{d}}$  and the modeled distances  $\mathbf{d}$  is minimized. This cost is stated in terms of displacements using Eq. 20

$$g(\mathbf{u}) = \frac{1}{2} \mathbf{u}^T \mathbf{Q} \mathbf{u} + \mathbf{f}^T \mathbf{u}, \quad (21)$$

where  $\mathbf{Q} = 2\mathbf{D}^T \mathbf{D}$  is symmetric positive-definite and  $\mathbf{f} = 2\mathbf{D}^T \mathbf{d}_0$ . The reader should note that an additional term  $\|\mathbf{d}_0\|^2$  is omitted from Eq. (21) since it is constant and would have no effect on the minimization process.

#### C. Linearization of the Ideal Gas Law

We note that the inverse problem formulation can be cast as an efficient-to-solve QP if the ideal gas law constraint imposed by Eq. (6) is linearized. This linearization together with Eq. (9) to write volume changes in terms of the displacements leads to the following linear constraint

$$p_v = -\frac{P_0}{3V_0} \mathbf{A}^T \mathbf{u}. \quad (22)$$

#### D. Inverse Problem and Contact Patch Estimation

The resulting inverse problem is formulated as the convex QP:

$$\begin{aligned} \arg \min_{\mathbf{u}=[\mathbf{u}, \mathbf{p}_c, p_v]} \quad & \frac{1}{2} \mathbf{u}^T \mathbf{Q} \mathbf{u} + \mathbf{f}^T \mathbf{u}, \\ \text{s.t.} \quad & \mathbf{K} \mathbf{u} + \text{diag}(\mathbf{A}) \mathbf{p}_c = \mathbf{A} \mathbf{p}_v, \\ & \frac{P_0}{3V_0} \mathbf{A}^T \mathbf{u} + p_v = 0, \\ & \mathbf{p}_c > \mathbf{0}. \end{aligned} \quad (23)$$

Note that this formulation finds an optimum solution in both  $\mathbf{u}$  and  $\mathbf{p}_c$  as well as in the internal pressure changes  $p_v$ . Therefore, the strategy has no knowledge of the geometry of the external object making contact with the membrane of the sensor. Once the solution is found, the contact patch is identified as those points of the mesh that have a non-zero contact pressure. However, with real data containing noise, the computed contact pressure exhibits spurious non-zero values at regions away from the contact patch. These values are significantly lower than those within the contact region and can be filtered by simple thresholding.

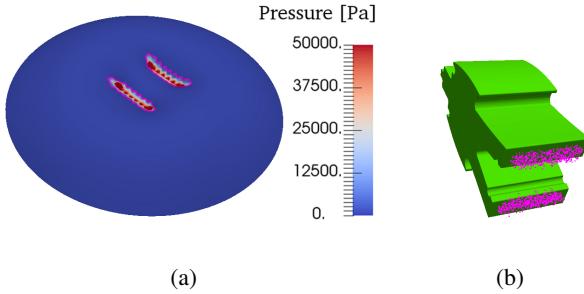


Fig. 4: Evaluation of the inverse problem strategy with the synthetic point cloud data generated by simulation and shown in Fig. 3b.(a) Deformed membrane shape predicted by solving the inverse problem. The surface is colored with the magnitude of the contact pressure; (b) Subset of the point cloud identified as corresponding to the contact patch.

#### E. Performance Against Synthetic Data

The performance of our inverse method tested against synthetic point cloud data generated for the simulation presented in Section III-C is shown in Fig. 3. Figure 4 shows the result from the contact patch detection strategy summarized in Eq. (23) for a given pose of the robot toy object. Estimated deformations are shown in Fig. 4a, colored by the estimated contact pressure. Note the low level of noise in the predicted deformations even though the point cloud data does contain noise. The computed contact pressure is interpolated onto each point of the cloud and thresholding is used to identify a subset that best corresponds to the contact patch. This subset is shown in Fig. 4b together with the original robot toy. Note that the methodology proposed is able to filter out the region between the legs of the robot toy even in the presence of high gradients and curvatures. Figure 4a shows the contour line (magenta) corresponding to the threshold level used to discriminate the contact patch. We chose a threshold level as a factor of the average contact pressure  $\mathbf{A}^T \mathbf{p}_c / \|\mathbf{A}\|_1$ .

#### F. Object pose estimation using contact patches

The proposed inverse method is ideally suited to be used within a perception pipeline for manipuland pose estimation. We combined the contact patch estimator with a dual-stage ICP based pose estimator as depicted in Figure 5. While [6] demonstrated an example of pose-estimation using ICP on the complete depth image, their approach is neither robust nor guaranteed to work for arbitrary manipuland geometries. By utilizing only the portion of the point-cloud corresponding to the contact patch in a first pass at computing ICP, a more robust initialization can be obtained for a subsequent refinement stage utilizing points in the local neighbourhood of the estimated contact patch. Our approach is faster than [6], since fewer points are utilized within the ICP stages, and is more robust since it better captures key object features, such as corners in the vicinity of the contact patch while rejecting portions of the bubble that are not in contact. An implementation detail relevant to the ToF sensor used within the *Soft-bubble* is that the raw point clouds captured are

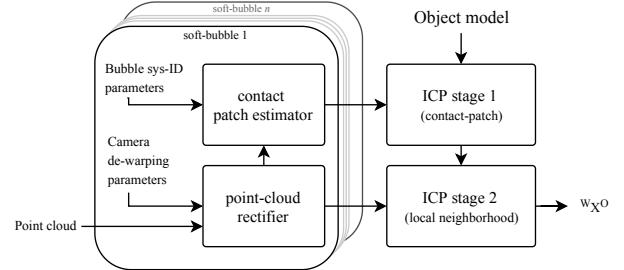


Fig. 5: Schematic of the perception pipeline for pose estimation using the contact patch estimator and a dual-stage ICP based pose estimator; the first stage utilizes points from the estimated contact patch and the second utilizes a local neighbourhood of points from the rectified point-cloud.

subject to warping due to close distances and a rectification stage is needed prior to subsequent processing using them.

## V. EXPERIMENTS

In this section we present various experiments validating our approach and demonstrating its efficacy in manipulation tasks.

#### A. Sensor and Robot Setup

We utilized *Soft-bubble* sensors [6] comprised of an elastic membrane sensing surface, an airtight hull allowing pressurization of the membrane, and an internal PMD *pico flexx* ToF based depth sensor. On inflation, the membrane forms a deformable spherical cap approximately 20 to 75mm in height. For the analysis presented here, we inflated it to a 50mm height with an internal pressure of 0.375psi. The complete sensor assembly weighed 500 grams and was mounted as the end-effector for the Kuka IIWA arms. The distortion observed in the ToF sensor at the 12 - 17cm depth was characterized using a simple linear per-pixel fit and then rectified. The camera was set to use fixed exposure and to stream data at 15Hz. The interface software for modelling and control utilized components of the Drake toolbox [23].

#### B. Soft-bubble System Identification

System identification for the *Soft-bubble* consists of fitting a single term,  $\gamma_0$  from Equation 2, using depth measurements of the inflated membrane in a contact-free state (since the reference pressure  $P_0$  is fixed *a priori* to 0.375 psi). We use the assumption that the bubble geometry matches a hemisphere and thus  $\gamma_0$  can be obtained from the analytic form of the Young-Laplace equation. To validate our system identification, we compare predicted and sensed point clouds from contact with a known object at a known pose. This was performed by using a Kuka IIWA arm to press the *Soft-bubble* onto a pyramidal frustum (similar to that shown in Figure 7) mounted at a known pose. The pose of the internal ToF depth sensor relative to the wrist of the robot is obtained from CAD models. Data capture and analysis for validation takes only a few minutes and is representative of the ease-of-use of our proposed method. Note that system identification is only dependent on reference pressure, sensor geometry

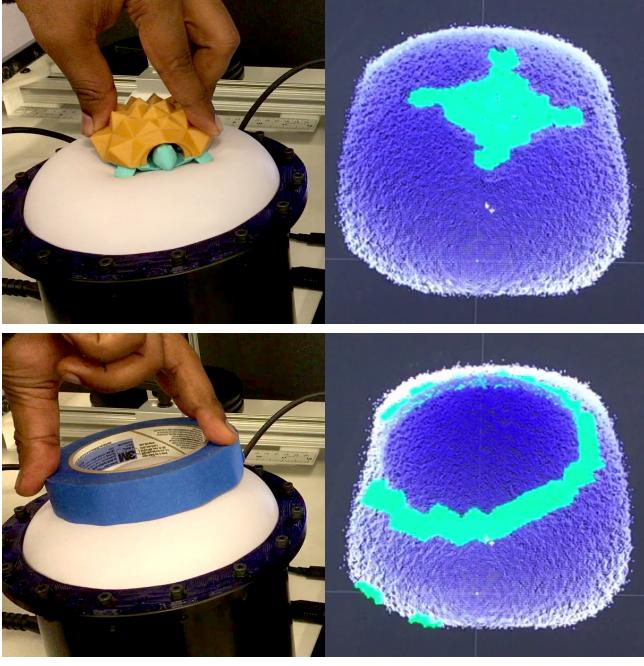


Fig. 6: Real-time contact patch estimation for two test objects pressed into the *Soft-Bubble* tactile sensor. The shape and size of the contact patch clearly reflects the contacting geometry.

(bubble diameter) and membrane material (latex) - as long as the chosen material has near uniform thickness; thus, the identified constants are identical for all additional sensors of the same form factor making it an extremely efficient and easy-to-deploy procedure.

### C. Contact-patch Estimation

Contact patch estimation was run asynchronously and computation time was measured at between 7-10Hz including overhead for pointcloud data serialization/deserialization when run on an average dual-core laptop. The contact patch estimator produces a contact pressure map  $p_c$  corresponding to each point on the measured point cloud. In practice, for our chosen depth sensor and its settings, this results in  $\sim 37,000$  points being processed for every frame captured. From the contact pressure map, the estimated contact patch was identified by a simple thresholding operation to eliminate spurious numerical results, where the threshold was set to  $\kappa \mathbf{A}^T \mathbf{p}_c / \|\mathbf{A}\|_1$ . In our experience, a scaling factor of  $\kappa = 1$  is sufficient in most cases. The computed contact patch was then published as a pointcloud to be consumed by the next stage of our perception pipeline.

Fig. 6 shows the computed contact patch for two objects pressed into the *Soft-bubble*. Note that the estimated contact patch closely captures the geometry of the contacting surface. Since our proposed method is agnostic to the object geometry, we observe that contact patches can be estimated for a wide variety of contacts that might be encountered in robot manipulation. Additional examples of contact patch estimation with a range of objects are included in the video accompanying this manuscript.

### D. Pose Estimation

Pose estimation of known reference objects using the dual-stage ICP approach was carried out utilizing the ICP implementation from PCL. Figure 7 shows the results of tracking the pose of a pyramidal frustum pressed into the sensor. The complete pipeline runs at approximately 2-5 Hz. While further analysis of the quality of the obtained pose-estimates are outside the scope of this paper, qualitatively, the computed object pose was observed to be a close-enough fit for dexterous manipulation tasks.

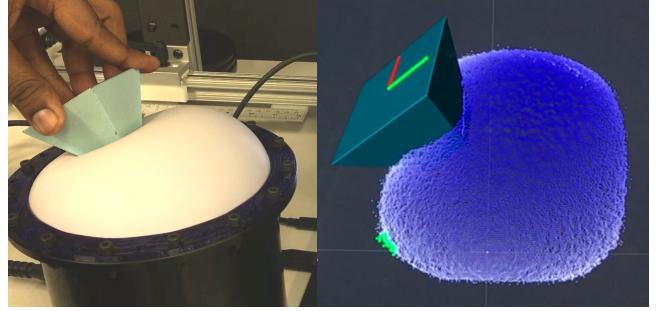


Fig. 7: Pose estimation of a pyramidal frustum shape using ICP integrated with our contact patch estimator. The sensed pointcloud is shown in blue.

### E. Dual finger manipulation

The *Soft-bubble* naturally lends itself to use as a compliant fingertip for manipulation tasks. We therefore mounted two *Soft-bubble* sensors on dual Kuka IIWA robot arms, as shown in Figure 1, as an analog to a pair of fingers. We performed an open-loop manipulation task in which the two arms screw a nut onto a bolt and used the *Soft-bubble* sensors to track the resulting contact patches. For the nut-turning task, arm joint configurations for turning the nut are computed using constrained optimization-based inverse kinematics and regrasp motions are generated using a bi-directional RRT planner [24]. The resulting contact patches from this rich contact manipulation were computed at a rate of 7-10Hz; Figure 1 shows a sequence of poses from the task with the corresponding sensed pointclouds and estimated contact patches. Not only do we accurately estimate the contact patch, the highly deformable membrane of the sensor provides a secure grasp on the nut. While our nut-turning task is currently performed open-loop, our *Soft-bubble* sensors provide feedback that could be used to accurately track the grasp on the nut and detect any unwanted slip. We plan to explore feedback control strategies using dense geometry feedback in future work.

## VI. CONCLUSIONS AND DISCUSSION

In this paper we have presented a model for the behaviour of highly-deformable dense-geometry sensors. While our *forward* model predicts deformation given the pose and geometry of a contacting rigid object, our solution to the converse *inverse problem* estimates the contact patch between

the sensor and arbitrary objects. We have shown that the latter, formulated as a sparse convex QP, can be efficiently computed and estimates contact patches with sufficient accuracy for use as the first stage in a perception pipeline for manipulation tasks. Used together with a dual-stage ICP-based pose estimator, it can accurately track the pose of objects contacting the sensor. Looking to the future, we have demonstrated the potential of these sensors using our methods for robotic manipulation on a contact-rich manipulation task - the sensors also act as end-effectors for over-sized fingers screwing a nut onto a bolt.

Feedback through contact is vital for performing robust manipulation. It can be used at short timescales as feedback to a stabilizing controller and at large timescales to monitor contact mode or type. A policy-based planning and execution approach like [25] exploits contact and robot compliance to perform manipulation tasks in the presence of uncertainty. For each action performed from the policy, the outcome of each action must be identified. For motions in contact, comparison between expected contacts from planning and observed contacts from a sensor like the *Soft-bubble* through our proposed contact-patch estimator is an ideal means for identifying the outcomes of actions.

Our proposed estimator is noteworthy for the speed at which it can be computed and the ease of system identification— a process that can be completed in a matter of minutes and generalizes well to other *Soft-bubble* sensors of the same materials and form-factor.

Our proposed approach is not without limitations; notably shear forces and friction are currently not estimated and this remains an open research question when utilizing these sensors. Therefore the current methodology cannot distinguish between similar geometries with different surface roughness. While linearizing around operating pressure significantly simplifies our model, it results in a sensitivity to the inevitable under- or over-inflation of the *Soft-bubble* sensor. We are currently investigating algorithmic and sensor design improvements that will mitigate these limitations. While we demonstrated a simple ICP-based tracker coupled with our contact-patch estimator, further work is needed for accurate and stable pose estimation allowing the integration of *Soft-bubble* sensors into a robust and complete perception pipeline for soft robotic manipulation.

## REFERENCES

- [1] C. Chi, X. Sun, N. Xue, T. Li, and C. Liu, “Recent progress in technologies for tactile sensors,” *Sensors*, vol. 18, no. 4, p. 948, 2018.
- [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, p. 2762, 2017.
- [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,” *arXiv preprint arXiv:1803.00628*, 2018.
- [4] A. Yamaguchi and C. G. Atkeson, “Implementing tactile behaviors using FingerVision,” in *Humanoids*, Nov 2017, pp. 241–248.
- [5] D. Ma, E. Donlon, S. Dong, and A. Rodriguez, “Dense tactile force distribution estimation using GelSlim and inverse FEM,” *arXiv preprint arXiv:1810.04621*, 2018.
- [6] A. Alspach, K. Hashimoto, N. Kuppuswamy, and R. Tedrake, “Soft-bubble: A highly compliant dense geometry tactile sensor for robot manipulation,” in *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*. IEEE, 2019, pp. 597–604.
- [7] L. Zou, C. Ge, Z. Wang, E. Cretu, and X. Li, “Novel tactile sensor technology and smart tactile sensing systems: A review,” *Sensors*, vol. 17, no. 11, p. 2653, 2017.
- [8] Z. Kappassov, J.-A. Corrales, and V. Perdereau, “Tactile sensing in dexterous robot hands review,” *Robotics and Autonomous Systems*, vol. 74, pp. 195 – 220, 2015.
- [9] S. Li, S. Lyu, J. Trinkle, and W. Burgard, “A comparative study of contact models for contact-aware state estimation,” in *IROS*, Sep. 2015, pp. 5059–5064.
- [10] M. Bauza, O. Canal, and A. Rodriguez, “Tactile mapping and localization from high-resolution tactile imprints,” *arXiv preprint arXiv:1904.10944*, 2019.
- [11] W. Yuan, R. Li, M. A. Srinivasan, and E. H. Adelson, “Measurement of shear and slip with a GelSight tactile sensor,” in *ICRA*. IEEE, 2015, pp. 304–311.
- [12] S. Dong and A. Rodriguez, “Tactile-based insertion for dense box-packing,” in *IROS*, 2019.
- [13] K. Takahashi and J. Tan, “Deep visuo-tactile learning: Estimation of tactile properties from images,” in *ICRA*. IEEE, 2019, pp. 8951–8957.
- [14] M. Polic, I. Krajacic, N. Lepora, and M. Orsag, “Convolutional autoencoder for feature extraction in tactile sensing,” *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3671–3678, 2019.
- [15] R. Calandra, A. Owens, D. Jayaraman, J. Lin, W. Yuan, J. Malik, E. H. Adelson, and S. Levine, “More than a feeling: Learning to grasp and regrasp using vision and touch,” *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3300–3307, Oct 2018.
- [16] 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 *CoRL*, 2017.
- [17] S. Tian, F. Ebert, D. Jayaraman, M. Mudigonda, C. Finn, R. Calandra, and S. Levine, “Manipulation by feel: Touch-based control with deep predictive models,” *arXiv preprint arXiv:1903.04128*, 2019.
- [18] R. A. Sauer, T. X. Duong, and C. J. Corbett, “A computational formulation for constrained solid and liquid membranes considering isogeometric finite elements,” *Computer Methods in Applied Mechanics and Engineering*, vol. 271, pp. 48–68, 2014.
- [19] R. L. Taylor, E. Oñate, and P.-A. Ubach, “Finite element analysis of membrane structures,” in *Textile Composites and Inflatable Structures*. Springer, 2005, pp. 47–68.
- [20] F. White, *Fluid Mechanics*. McGraw-Hill Higher Education, 2015.
- [21] T. J. Hughes, *The finite element method: linear static and dynamic finite element analysis*. Courier Corporation, 2012.
- [22] O. Sorkine and D. Cohen-Or, “Least-squares meshes,” in *Shape Modeling Applications*. IEEE, 2004, pp. 191–199.
- [23] R. Tedrake and the Drake Development Team, “Drake: Model-based design and verification for robotics,” 2019. [Online]. Available: <https://drake.mit.edu>
- [24] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.
- [25] C. Phillips-Grafflin and D. Berenson, “Planning and resilient execution of policies for manipulation in contact with actuation uncertainty,” in *WAFR*, 2016.