Soft, round, high resolution tactile fingertip sensors for dexterous robotic manipulation

by

Branden Robert Romero

B.S., University of Colorado Boulder (2017)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2022

Author .....
Department of Electrical Engineering and Computer Science
August 26, 2022

Certified by .....
Edward H. Adelson
John and Dorothy Wilson Professor of Vision Science
Thesis Supervisor

Accepted by .....
Leslie A. Kolodziejski
Professor of Electical Engineering and Computers Science
Chair, Department Committee on Graduate Students

Soft, round, high resolution tactile fingertip sensors for dexterous robotic manipulation

by

Branden Robert Romero

Submitted to the Department of Electrical Engineering and Computer Science
on August 26, 2022, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

In this work we introduce a non-planar soft high-resolution tactile sensor. An iteration of the GelSight sensors, it enables future GelSights to have more complicated form factors, such as a humanoid fingertip. To do this we introduce a novel method for achieving directional lighting along the entirety of a curved sensor using light piping. Light piping uses total internal reflection and a semi-specular membrane to constrain the path of the light inside the sensor until the sensing membrane is deformed. By using this new membrane and changing the geometry, we introduce a new bidirectional reflectance distribution function and new optics. This require new calibration procedures in the form of developing a fisheye projection model, and developing a neighborhood and location based continuous look-up table to map the relationship between RGB value and surface normal orientation of the membrane at a point. Finally we perform two dexterous manipulation task with feedback from the sensors in the form of controlled rolling of an object on a support surface, and lid removal off a jar. We also give instructions on how to manufacture the sensor as well as increasing the durability of the membrane for all GelSight sensors.

Thesis Supervisor: Edward H. Adelson
Title: John and Dorothy Wilson Professor of Vision Science

Acknowledgments

I would like to first thank Kristi and Ryan Yoho for their support throughout my life.

Next up, I would like to say it has been a unique and wonderful opportunity to work with Prof. Edward Adelson. It is quite inspiring to work with him and it is quite inspiring to see someone so dedicated to their work and their sense of esthetics.

I would also like to thank Prof. Nikolaus Correll for introducing me to the world of robotics, and trusting me even though I set one of your robots on fire.

Lastly thank you for all the support from my lab mates and all visting students, Shaoxiong Wang, Sandra Liu, Megha Tippur, Radhen Patel, Yu She, Filipe Veiga, Chen Wang, Achu Wilson, Peiyu Yu, Satoshi Funabashi, and Emma Hartman.

Contents

1 Introduction **15**

2 Related Work **17**

2.1 Tactile Sensing 17

2.1.1 Vision-Based Tactile Sensing 18

2.2 Application of Vision-Based Tactile Sensors 19

3 Design **21**

3.0.1 Motivation 21

3.1 Light Piping 22

4 Manufacturing **25**

4.1 Mold Making 25

4.2 Casting 26

4.3 Assembly 27

5 Software and Calibration **29**

5.1 Software Interface 29

5.2 Sensor calibration 29

5.3 2D-3D Correspondence 30

5.3.1 Experimental Setup 30

5.3.2 Method 31

5.3.3 Results 33

5.4 RGB to Gradient Correspondence 34

5.4.1 Experimental Setup . . . 35
5.4.2 Method . . . 36
5.4.3 Results . . . 36
5.5 3D Reconstruction . . . 37

6 Controlled Rolling **39**
6.1 Method . . . 40
6.1.1 Tracking Contact Area . . . 40
6.1.2 Controller . . . 40
6.2 Results . . . 41

7 Cap Removal **43**
7.1 Experimental Setup . . . 43
7.2 Method . . . 43
7.2.1 Localization . . . 44
7.2.2 Relocation . . . 44
7.2.3 Unscrewing . . . 45
7.2.4 Determining if Unscrewed . . . 45
7.3 Results . . . 45

8 Conclusion **47**
8.1 Future Work . . . 48
8.1.1 Hardware . . . 48
8.1.2 Software . . . 48

List of Figures

1-1 An Allegro Hand equipped with four sensors holding a mesh cup. Each finger provides a high resolution 3D reconstruction of its respective contact areas. 16

3-1 Human demonstration illustrating rolling a cube along the frontal plane of the finger. (a) the rolling motion of a curved finger which maintains a constant contact area size and has a wider range of motion. (b) shows the rolling motion with a flat finger which has a shrinking contact area and limited range of motion. 22

3-2 Human demonstration illustrating contacts made on different parts of a cylinder using the finger's transverse plane. (a) the contact areas made with a curved finger,(b) the contact areas made with a flat finger. The round finger keeps a consistent contact area at each contact point unlike the flat finger. 23

3-3 CAD model of the sensor illustrating the assembled finger, along with the exploded view of the sensor showing the internal components. . . 24

3-4 Illustration showing the light piping illumination system along a single axis. (a) the path in which the light travels without contact. (b) the path of the light when contact is made. 24

5-1 Illustration showing the projection model used for our fisheye lens. . . 31

5-2 **Left** shows the surface being observed by the fisheye lens. **Right** shows the image projected onto an explicit 3D representation of the surface in the form of a point cloud. Due to some error in the projection model some points outside of the sensing region are projected onto the point cloud and hence the black points. 33

5-3 Ball being pressed into **From left to right** GelSight sensor equipped with a semi-specular membrane and a lambertian membrane. Illustrates that, compared to lambertian surface, the semi-specular lacks a one-to-one correspondence between surface normal orientation and observed RGB value. Illuminated by four LEDs, the semi-specular membrane has almost constant color for regions reflecting a single LED color. Meanwhile, the lambertian surface illuminated by three LEDs can allow you to easily tell the direction of illumination as it blends smoothly from one region illuminated by one color to another. . . . 34

5-4 Shows the issue of non-uniform illumination. As the deformation occurs further away from the illumination source, the more significant the perceived brightness of that source. Here, while the left and bottom of the ball are well illuminated, the illumination on the right of the ball which should be blue is instead seen as a shadow. 35

5-5 Illustration of network architecture for estimating surface normal orientation from RGB pixel value based on pixel location and neighborhood information. 36

6-1 Sample begin and end state of each object during the rolling phase of the experiment in the controlled rolling task. 39

6-2 **Top from left to right** Sequence of the experiment being performed on the plastic sphere during the rolling stage. **Bottom from left to right** The corresponding point clouds showing the evolution of the point cloud as the object rolls. 40

6-3 Illustration showing the experimental set-up. **From left to right** The placement of the object, when the object makes contact with both fingers, the end state of the object when the object reaches the desired position. 41

7-1 Shows the process of cap removal. **From left to right and top to bottom** An object is randomly placed by a human in the workspace of the manipulator. The robot then uses tactile search to localize the jar. Once reaching the ideal location of the jar the robot sweeps it into a position more suitable for unscrewing. The robot then continuously unscrews the jar. Once it detects the jar lid is no longer moving up it removes the lid. The procedure takes less than 11 seconds. 44

List of Tables

5.1 The RMSE of the test set on our four fisheye projection models. Here the error is the error between the estimated image plane projection vs the ground truth colleted from an OptiTrack and ChAruco system. . 33

5.2 Shows the MSE of including or not including the pixel location when estimating surface normal orientation from RGB pixel value. 37

5.3 Shows the influence of aggregating neighborhood information while estimating surface normal orientation from RGB pixel value. In particular the affect of having different window sizes. 37

Chapter 1

Introduction

With the proliferation of vision-based tactile sensors [15, 35, 33, 26, 12] we have seen major advancements in the field of robotics [2, 14, 36, 25, 29, 8, 32, 30, 37, 22, 34, 31]. However impressive this may be, intelligent behavior extends beyond the brains of any system. These systems obviously benefit from having high-resolution information about the contact interaction happening at the finger tips, but they are still limited by the capabilities of their manipulator, which happens to be in most cases a parallel gripper. As a result, we see behaviours perhaps too slow, behaviours that are too conservative and rely too much careful planning, or behaviours that are extremely risky and perform dynamic movements to overcome the limitations of their bodies. While it is easy to say that we just need to increase the complexity of our manipulators, decades of robotics research says that its not that straight forward, dealing with high degrees-of-freedom with complex contact dynamics is beyond the capabilities of most methods. With the continuing rise of machine learning, perhaps these issues can be addressed. So, in this work we develop a platform that allows users to get the same rich contact information, while not inhibiting the capabilities of their dexterous robotic manipulators. In this work we introduce a new variant of the GelSight sensors that enables new finger geometries to be developed besides a planar surface. To this end we introduce a new optical design inspired by wave optical fibers, to achieve directional illumination along the entirety of a curved surface in Chapter 3. We then show how to manufacture such a design via a gravity-fed molding sensing

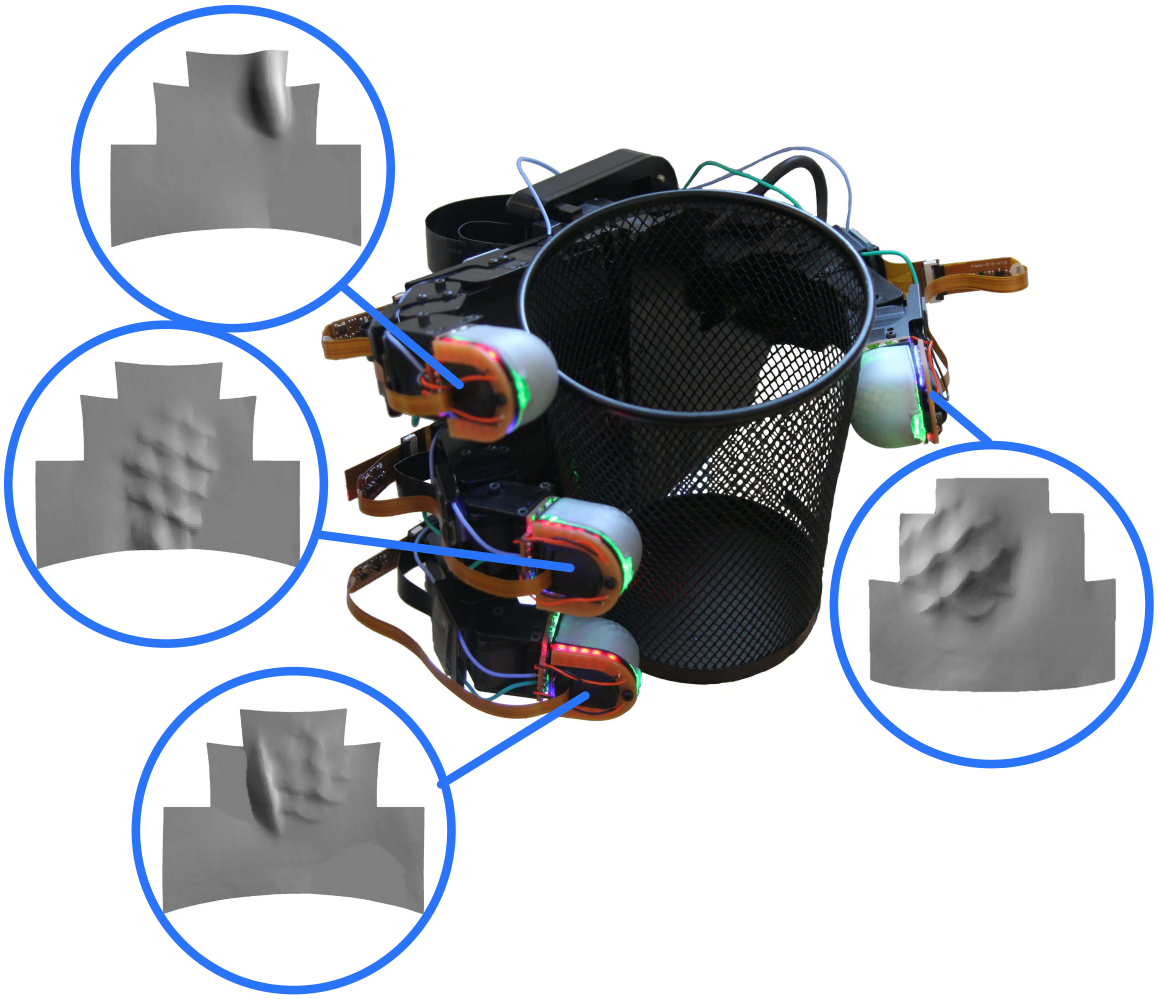

Figure 1-1: An Allegro Hand equipped with four sensors holding a mesh cup. Each finger provides a high resolution 3D reconstruction of its respective contact areas.

while providing methods for increasing the durability of the sensors membrane in Chapter 4. Due to changes in the bidirectional reflectance distribution function of the surface compared to previous sensors as well as a change in optics to a fisheye lens we introduce a new calibration procedure in Chapter 5. Finally, to show the capabilities of these sensors, we perform two task seen in daily life opening a jar and controlled rolling of an object on a surface to reorient it for grasping in Chapter 6 and Chapter 7.

Chapter 2

Related Work

Observations from [10] and [5] illustrate that tactile sensing and softness play an important role in dexterous manipulation. Where the former illustrates that even with the intelligence of a human it is difficult to perform precise dexterous manipulation without tactile sensing, as participants with anesthesia injected into their fingertips fumble objects. Meanwhile, the latter shows that a robotic hands made out of soft materials passively comply to object shapes to stabilize forces, making finger control simpler.

2.1 Tactile Sensing

Inspired by these finding a plethora of tactile sensors were developed. Soft sensors like the SynTouch BioTac and [19] sought to mimic the signals from mechanoreceptors of human skin in the fingertips [11]. Where the BioTac sensor is a tactile sensor with an elastic skin filled with an incompressible conductive fluid, that uses thermistor, impedance sensing electrodes, and hydro-acoustic pressure sensors to sense normal and tangential forces along with the thermal conductivity of objects in contact. [19] is a proximity, contact, and force sensor (PCF sensor) that uses an infrared proximity sensor embedded in a transparent elastomer to estimate the proximity of an object to the finger, as well as force applied to the finger tip. Apart from these sensors various types of other tactile sensors based of transduction like, strain gauges, resis-

tive/piezoresistive, ultrasound, tunnel effect, impedance, piezoelectric and magnetic sensors can be found in the following review [4].

2.1.1 Vision-Based Tactile Sensing

However, these sensors were low resolution compared to the resolution of the sensing found in the human finger tips. Rather than reducing the size of electrical components, vision based tactile sensors use a remote high-resolution camera sensor to observe a contact surface. Paired with their softness vision-based tactile sensors have become very prominent [15, 35, 33, 26, 12]. The Soft-Bubble Gripper [12], is an air-filled latex membrane observed by a ToF depth sensor to get the 3D membrane deformation. In [27] dots are randomly dispersed in an opaque membrane and use optical flow to estimate the force distribution being applied to the membrane. The TacTip sensors [20] have been designed in various form factors and use a vision sensor to observe dots printed on a soft 3D printed membrane, and use the change in dot location when the membrane is deformed to reconstruct the contact patch. Similar to TacTip, FingerVision [33] uses a clear membrane with dots placed on it, but instead uses it to estimate the forces and torques being applied to the finger. GelSight sensors [35] use the camera to observe a clear elastomer covered in a reflective membrane. The membrane is directionally illuminated which allows 3D reconstruction of the surface using photometric stereo. Dots have also been placed on the surface to detect slip and tangential forces [35]. There are a variety of sensors derived from Gelsight [13, 18, 6, 9], but usually omit the photometric stereo, and work from raw RGB image instead. Most noteably, [9], uses the principles of a two-way mirror, and adjusts the illumination in the sensor to make the reflective membrane transparent or opaque, to see an object prior to grasping, and [18] uses an endoscope camera array to create a curved finger to get a 360 degree view of the interior of a curved sensor.

2.2 Application of Vision-Based Tactile Sensors

The class of vision-based tactile sensors have enabled many robotic manipulation tasks. Broadly speaking it has been used for either system identification, i.e shape geometry and dynamic properties, or for control. In terms of control, vision-based tactiles sensors have been used for tasks such as learning grasp success [2], servoing object surfaces [14], detecting slip and shear force [36], object insertion [25], cable manipulation [29], and object reposing [8]. While in terms of system identification it has been used in reconstructing 3D surfaces [32, 30], distinguishing between different cloth materials [37], estimating object pose [1], finding objects in granular media [22], determining material softness [34], and learning physical features to perform dynamic in-hand manipulation [31].

Chapter 3

Design

3.0.1 Motivation

Consider the task of flipping an object that is resting on a table, as depicted in Fig. 3-1. Here the index finger of a hand rolls an object that is lying on a supporting surface towards the thumb. Of the two cases depicted, we can see that when using flat sensors, the contact patch location and size greatly vary throughout the manipulation trajectory, with the size of the contact patch being reduced to almost a point contact when it reaches the edge of the sensor. Having such a small contact patch not only reduces the stability of the object, but also reduces the robots perception of the objects state. On the other hand, when a curved sensor is used, while the location of contact still changes in a similar manner, the contact patch size remains relatively consistent throughout the object trajectory. This decrease in variation of the contact patch size makes it much easier to track the object state.

Another case where fingertip shape clearly impacts the performance of dexterous hand systems is when performing grasp quality assessment. For assessing the quality of a grasp it is critical that the contact areas acquired after grasping the object are perceivable by the fingertip sensors. When using flat sensors, in order to maximize the contact patch information, the fingers have to be reoriented such that the sensing surface is orthogonal to the contact location. As previously stated, this can be problematic when considering the limited kinematic structure of each finger. Since

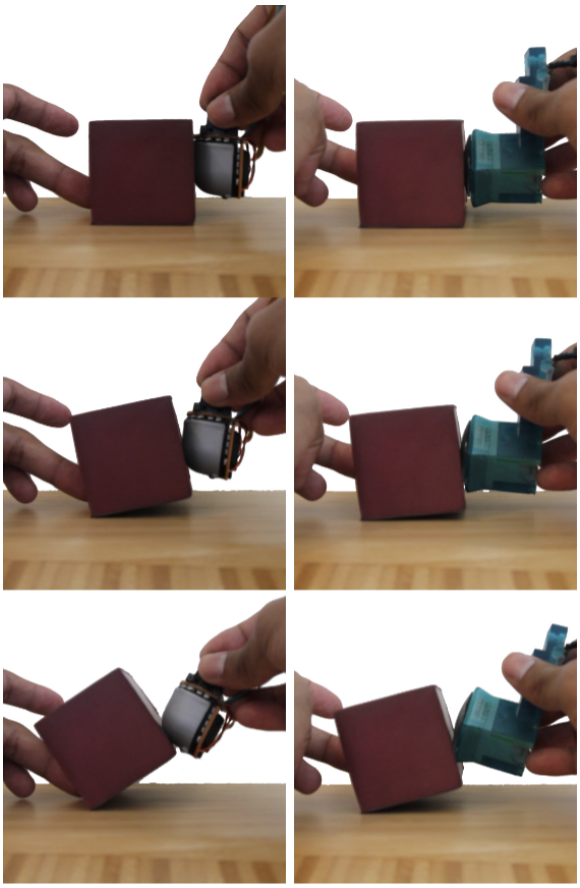

Figure 3-1: Human demonstration illustrating rolling a cube along the frontal plane of the finger. (a) the rolling motion of a curved finger which maintains a constant contact area size and has a wider range of motion. (b) shows the rolling motion with a flat finger which has a shrinking contact area and limited range of motion.

curved fingertips are able to perceive contact patches in a wider ranged of orientations, explicitly reorienting the fingertip becomes unnecessary. An example of a grasp configuration where the differences between the two sensors are visible is also depicted in Fig. 3-2.

3.1 Light Piping

The goal of our design (Fig. 3-3) is to enable robots with dexterous manipulators to have rich information about the contact, in particular dense information about the 3D geometry of the contact areas. We propose a novel illumination system, that despite more complex geometry, allows us to perform 3D reconstruction despite some

Figure 3-2: Human demonstration illustrating contacts made on different parts of a cylinder using the finger's transverse plane. (a) the contact areas made with a curved finger,(b) the contact areas made with a flat finger. The round finger keeps a consistent contact area at each contact point unlike the flat finger.

limitations.

To achieve this design we make sacrifices in terms of illumination quality as seen in the previous sensor [35] hoping that the reconstructed height map is still suitable for robotic manipulation. In particular we use an opaque semi-specular sensing surface rather than a lambertian sensing surface. This results in an ambiguous surface normal orientation, since the RGB value does not map one-to-one to the surface normal of the sensing surface at a single location.

These choices were made in order to provide as uniform of an illumination pattern as possible along the entire surface of the finger. To do this we relied on a technique called light piping. Light piping is inspired by fiber optics in which a light is constrained within a medium via total internal reflection (TIR). We achieve something similar by using a semi-specular sensing surface and a thin plastic shell. We design the sensor in this manner, because semi-specular surfaces only slightly diffuses light while the lambertian surface used in the previous sensor completely diffuses the light. This means that rather than the light dissipating as it travels along the sensor

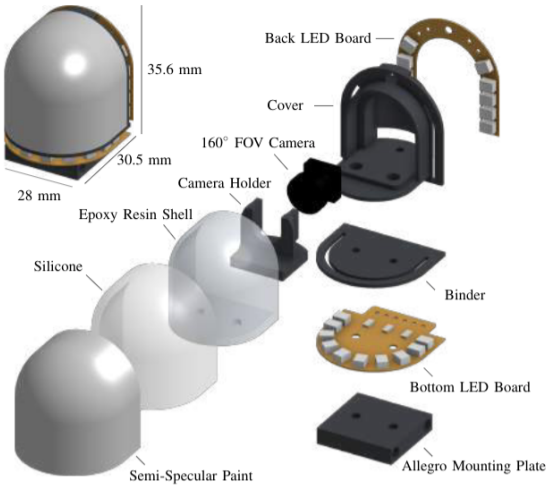

Figure 3-3: CAD model of the sensor illustrating the assembled finger, along with the exploded view of the sensor showing the internal components.

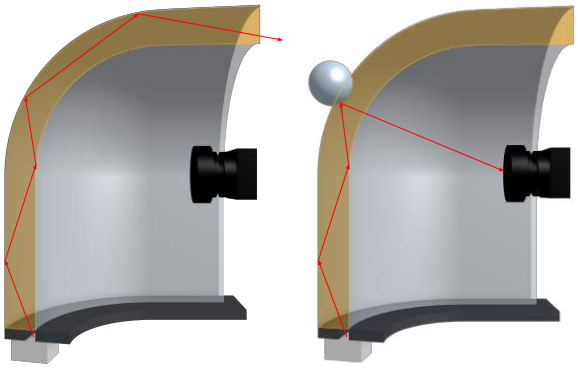

Figure 3-4: Illustration showing the light piping illumination system along a single axis. (a) the path in which the light travels without contact. (b) the path of the light when contact is made.

like it would with a lambertian surface, it will keep its directionality and thus more uniformly illuminate the surfaces beyond the curve of the surface Fig. 3-4. The thin plastic shell keeps the light from escaping into the interior of the sensor via TIR unless contact is made.

Chapter 4

Manufacturing

As a consequence of a more complicated geometry we have a more complicated manufacturing process compared to previous versions of the GelSight sensors. Where the previous sensor only required 3D printing, laser cutting, casting silicone into an open face mold, and pour over methods for coating, we rely on manufacturing a series of two-piece molds, casting into said molds, and then applying the opaque coating using an airbrush. However, in this process we have created methods that also significantly increase the durability and reliability of the sensors compared to the previous version.

4.1 Mold Making

First, the desired geometry of the plastic shell, as well as the geometry of the silicone and the shell combined, are 3D printed with the Form Labs Form 2 SLA printer using the clear resin. The cast pieces have to be optically clear, so we corona treat the surface of our 3D printed pieces. This is to increase the material's surface tension, which increases the adhesive properties of the print. We then dip coat each piece in an epoxy resin, Smooth-On XTC-3D, diluted with denatured alcohol (12 parts resin to 1 part of denatured alcohol by mass). This is allowed to dry at room temperature for 12 hours. This is done multiple times until the print no longer has layer lines or 3D printing artifacts. The reference plastic shell piece is then used to create a two-piece silicone mold. We chose Smooth-On MoldMax XLS II as our mold material because

we cast epoxy resin into these molds, and we found this material robust to multiple castings compared to other mold materials we used.

The second mold will be used to cast silicone onto the plastic shell, so we create a two piece mold where the base of the mold is a rigid 3D printed piece that will be rigidly attached to the shell, and the other piece is a soft silicone mold made from Smooth-On MoldStar 20T.

4.2 Casting

We begin the process by casting the plastic shell. Here we chose to cast the shell with a clear epoxy resin, in particular Smooth-On Epoxacast 690. The shell is only 1mm thick so we chose this material because of its low-viscosity, clarity, rigidity, and overall ease of use. After casting the shell we let the piece sit for 24 hours to completely cure. The next part tries to address some of the limitations of the previous sensor in terms of durability. In particular, in the previous sensor the paint was easy to remove and the gel easily delaminated from acrylic window. These issues stem from the fact that, one it is difficult to get silicone to stick to anything, and secondly that things were mechanically attached rather than chemically. To start we begin by priming the surface of our plastic shell with Dow DOWSIL P5200. This promotes silicone's adhesion to a variety of surfaces, but the caveat is the silicone must cure on that surface to form a chemical bond, rather than mechanically attaching a cured piece of silicone to a surface like in the previous Gelsights. The next part is creating the opaque coating. After experimenting with a variety of coatings we decided on creating a custom coating that is quite durable. The coating is made out of a silicone paint base, Smooth-on Psycho Paint, and a non-leafing silver dollar aluminum flake pigment. We spray this coating, so we dilute it with a silicone solvent, Smooth-On NOVOCS. The exact ratio used is 1:10:30 pigment, silicone paint base, and silicone solvent ratio by mass. We spray the interior of the silicone part of our mold with Mann Release Technologies Ease Release 200, and then spray our opaque coating in with an airbrush. We quickly screw our plastic shell onto our mold base, assemble the

mold, and pour our optically clear silicone gel (Silicone Inc. XPS-565 1:15 A:B ratio by mass) into the mold. We want to cast the silicone before the coating cures so that they are chemically bonded. The mold is left out for 6 hours at room temperature and then is placed in the oven at 95 degrees Celsius.

4.3 Assembly

The camera holder, cover, blinder, and mounting plate are 3D printed on the Markforged Onyx One printer with the Onyx filament which is very suitable for creating strong fixtures. The camera, Raspberry Pi Camera V2, is then press fitted into the camera holder. The camera has a high FOV of 220 degrees which observes majority of the sensing surface, while being significantly more compact than previous cameras used in GelSight. The camera holder is then press fitted into the cover and then screwed in with a M2 screw.The back LED board is screwed into the cover using an M2 screw. The cover is then press fit into the plastic shell. Two M2 screws are inserted into the bottom of the Allegro mounting plate that then pass through the through holes of the bottom LED board, blinder, and plastic shell and then screwed into the cover. The two LED boards are soldered together via four wires. Two power cables are routed to a Raspberry Pi Zero to power the LEDs with 3.3V, and then the camera is connected to a Raspberry Pi via CSI flat connector cable.

Chapter 5

Software and Calibration

5.1 Software Interface

The image from the sensor is streamed via HTTP. The image is streamed 640x480 with a frame rate of 90FPS. The latency is low with a measured latency of about 40ms delay. So far no computation happens on the Raspberry Pi. In terms of interfacing all four sensors with a host computer, each camera is connected to a Raspberry Pi Zero single board computer (SBC) using the standard CSI flat connector cables, then the four Raspberry Pis are then connected to a gigabit Ethernet switch in which the host computer is also connected to. In previous experiments, we tried interfaceing all cameras into a single (SBC), in particular the Jetson TX2, however it was determined to not be suitable for the application. While computation could be handled on the Jetson TX2 via its integrated GPU, the CPU-GPU data marshalling added an additional 110ms of latency to the camera stream, and required the images to be down-sampled to 160x120.

5.2 Sensor calibration

The calibration process sets out to solve two problems. The first is to find the 2D-3D correspondence in the form of pixel to point in point cloud; the second is to map the RGB value at a pixel to the surface orientation at that location (Sec. 5.4).

In terms of mapping RGB values to gradient we can no longer use a single look-up table like the previous GelSight sensors [35]. Despite trying to achieve uniform lighting throughout the whole sensing surface there are obvious non-uniformities, so in previous work [23] we proposed constructing a look-up table for a set of regions. Once the look-up tables were constructed and we perform 3D reconstruction, the methods know nothing about the curvature of the finger so we proposed a piece-wise forward projection of our height map onto the geometry of the surface. However, the discretization led to artifacts on the boundary of each region if deformation was large. In this work we propose a continuous look-up table in the form of a neural network. Additionally, in the previous work only a portion of the sensing surface was observable, so in this work we replace the 160 degree lens with a 220 degree lens. This adds additional challenges, since the lens causes large distortion that is difficult to model. So, in this work we also benchmark a set of fisheye projection models (Sec. 5.3).

5.3 2D-3D Correspondence

We start by determining the projection model of our lens. This procedure involves setting up a tracking system to track a 3D target, as well as a system to accurately determine projected location onto our image sensor from key points located on our target. A projection model is then determined by minimizing the estimated key point pixel location with the ground truth key point pixel location. Due to the uncertainty of the model behind the lens we benchmark several models.

5.3.1 Experimental Setup

We first create a target. We choose to use a variant of the Aruco markers [24], Charuco markers, which are fiducials embedded in a chessboard. This allows us to get accurate sub-pixel key point location due to the chessboard, while also being able to uniquely identify each chessboard square corner. We mount retroreflective markers on our target so that we can track its 3D pose accurately with an OptiTrack

(OptiTrack, NaturalPoint Inc.) system. The camera is removed from the sensor and placed rigidly to a fixture that is localized with the OptiTrack system. We then collect 600 frames of the target being observed by the camera, which gives us approximately 10800 pixel to 3D point correspondences.

5.3.2 Method

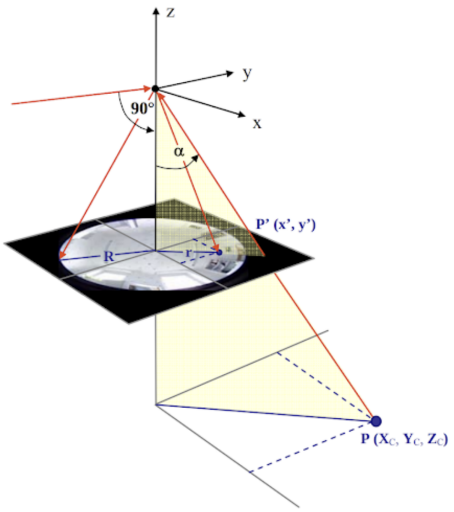

Figure 5-1: Illustration showing the projection model used for our fisheye lens.

Once the correspondences are collected, we have to determine which fisheye model is the most suitable model for our lens. So, after empirically determining that the lens follows the equidistant fisheye lens model, we determined that the fisheye lens has the following projection model (Fig. 5-1):

$$x' = \frac{\frac{2R}{\pi} tan^{-1}(\frac{\sqrt{X_C^2+Y_C^2}}{Z_C})}{\sqrt{\frac{Y_C^2}{X_C^2}}} + dx + x_h$$

$$y' = \frac{\frac{2R}{\pi} tan^{-1}(\frac{\sqrt{X_C^2+Y_C^2}}{Z_C})}{\sqrt{\frac{X_C^2}{Y_C^2}}} + dy + y_h$$

where, (dx, dy) are the distortion compensation functions, $(x', y'$ are the image coordinates of the projected points, (X_C, Y_C, Z_C) are the points in 3D, R is the fisheye

radius, and (x_h, y_h) is the coordinates of the principle point.

Since the distortion model is unknown for this lens, we benchmark the following distortion models:

1. Model 1 [7]

$$dx = x'(A_1 r'^2 + A_2 r'^4 A_3 r'^6 + A_4 r'^8)$$

$$dy = y'(A_1 r'^2 + A_2 r'^4 A_3 r'^6 + A_4 r'^8)$$

2. Model 2 [7]

$$dx = x'(A_1 r'^2 + A_2 r'^4 A_3 r'^6) + B_1(r'^2 + 2x'^2) + 2B_2 x'y' + C_1 x' + C_2 y'$$

$$dy = y'(A_1 r'^2 + A_2 r'^4 A_3 r'^6) + 2B_1 x'y' + B_2(r'^2 + 2y'^2)$$

3. Model 3

$$[dx, dy] = f_1^\theta(x', y', r')$$

4. Model 4

$$[dx, dy] = f_2^\theta(X_C, Y_C, Z_C)$$

where, $r' = \sqrt{x'^2 + y'^2}$, (A_1, A_2, A_3) are the radial distortion parameters, (B_1, B_2) are the decentering distortion parameters, and C_1, C_2 are the horizontal scale and shear factors.

To determine the parameters of the Model 1 and Model 2 [7] we use differentiable programming and perform gradient descent. Meanwhile, the parametric models are represented as multilayer perceptrons (MLP). Since, gradient descent is likely to converge at a solution at a local minima, we optimize on 50 random parameter initializations for each model trained with differentiable programming, and choose the parameters with the lowest error.

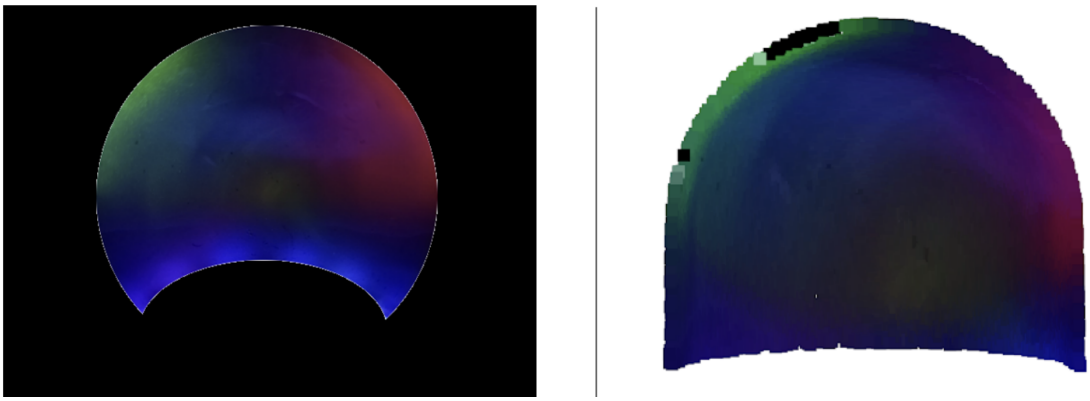

Figure 5-2: **Left** shows the surface being observed by the fisheye lens. **Right** shows the image projected onto an explicit 3D representation of the surface in the form of a point cloud. Due to some error in the projection model some points outside of the sensing region are projected onto the point cloud and hence the black points.

5.3.3 Results

After optimizing/training these models we get the following normalized RMSE error (Table 5.1):

Model 1	Model 2	Model 3	Model 4
1.0	0.945	0.324	0.155

Table 5.1: The RMSE of the test set on our four fisheye projection models. Here the error is the error between the estimated image plane projection vs the ground truth colleted from an OptiTrack and ChAruco system.

Due to its performance we choose Model 4 as our model for future projections. Since, Model 4 is a neural network, and generalization is of concern we modify our model so that it no longer takes (X_C, Y_C, Z_C), but instead the normal vector between (X_C, Y_C, Z_C) and our sensors optical center, since in our model any set of points that share the same normal between them and the optical center project to the same location on the image plane.

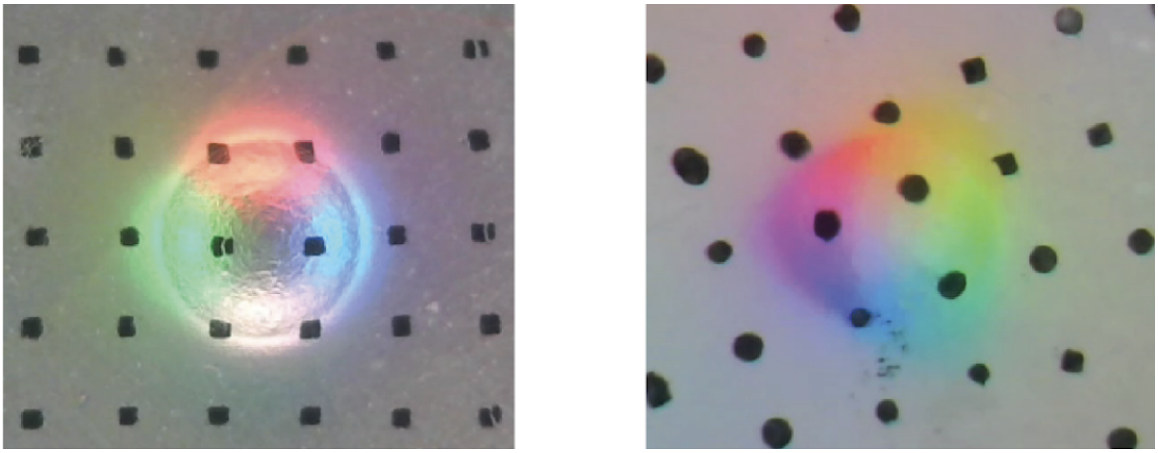

Figure 5-3: Ball being pressed into **From left to right** GelSight sensor equipped with a semi-specular membrane and a lambertian membrane. Illustrates that, compared to lambertian surface, the semi-specular lacks a one-to-one correspondence between surface normal orientation and observed RGB value. Illuminated by four LEDs, the semi-specular membrane has almost constant color for regions reflecting a single LED color. Meanwhile, the lambertian surface illuminated by three LEDs can allow you to easily tell the direction of illumination as it blends smoothly from one region illuminated by one color to another.

5.4 RGB to Gradient Correspondence

Now that we have a model for the projection model, we need to determine how the RGB value at a pixel changes as the sensing surface is deformed. In [35] this was done by creating a global look-up table that maps RGB values to surface normals. However, due to design trade-offs we encounter two issues. The first issue is that while light-piping allows light to travel along the curve of the sensing surface it comes at a cost that by using a semi-specular surface instead of a specular surface, each time a light ray reflects off the surface it has the potential to scatter in a random direction due to the micro-geometry of the surface. This causes the light observed further away from the light source to be significantly dimmer than if it would be if it was closer (Fig. 5-4). Secondly, the semi-specular surface reflects light in such a way that it is no longer possible to uniquely determine the surface normal as it was in previous sensors using a lambertian membrane (Fig. 5-3). So, we propose a continuous neighborhood based lookup table. In this section we will cover the experimental setup, the methods used, and benchmark how neighborhood size influences the performance of our surface

normal estimates.

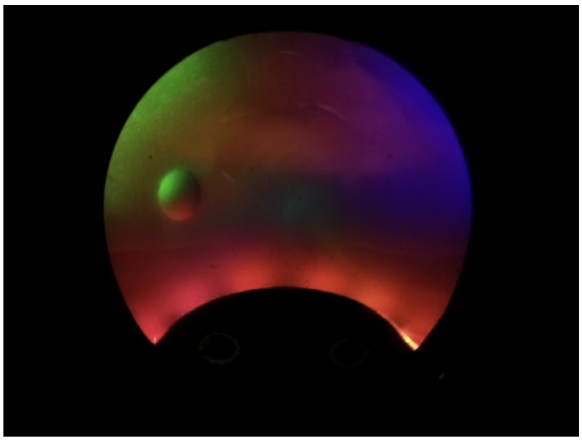

Figure 5-4: Shows the issue of non-uniform illumination. As the deformation occurs further away from the illumination source, the more significant the perceived brightness of that source. Here, while the left and bottom of the ball are well illuminated, the illumination on the right of the ball which should be blue is instead seen as a shadow.

5.4.1 Experimental Setup

We use the projection model from the previous section to determine the RGB to surface normal correspondence. We start by reassembling the sensor. We place the sensor on a rigid fixture with retro-reflective markers. We also created a probe composed of a 5/8" caster ball attached to a fixture also fitted with retro-reflective markers. Upon data collection, the sensor and the probe are localized with the OptiTrack system. The probe is then rolled along the surface of the sensor, ideally until each point on the surface has been deformed makes contact with each point along the ball of the probe. At each time step an image is collected, and since we have a CAD model of both the probe and sensor, we calculate which part of the probe is in contact with the sensing surface. We sample the points of the probe in contact with the sensing surface, calculate their surface normals and then project those points onto the image plane to see what the corresponding RGB value is. For this experiment we collected 18,000 frames over the course of 6 minutes.

5.4.2 Method

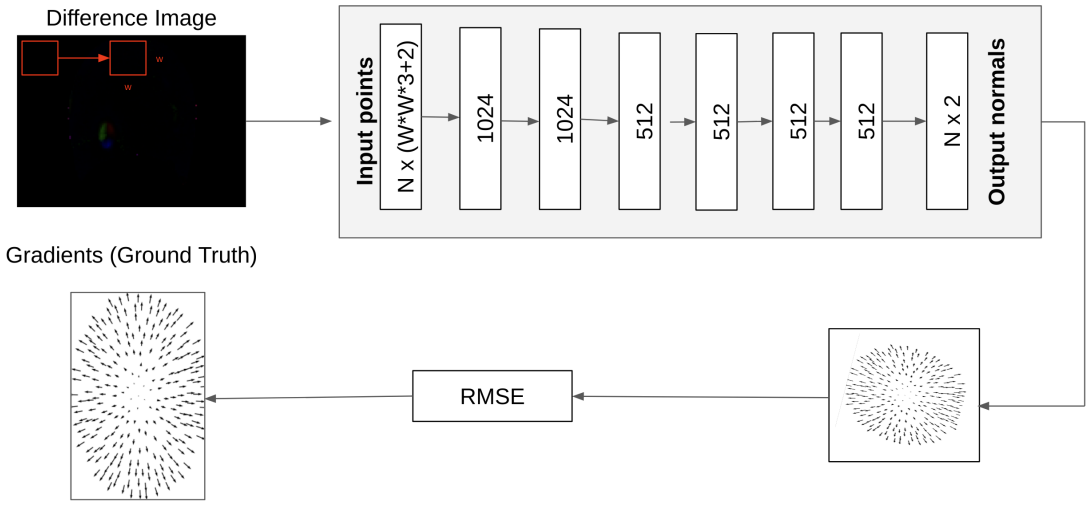

Figure 5-5: Illustration of network architecture for estimating surface normal orientation from RGB pixel value based on pixel location and neighborhood information.

To overcome the issues of non-uniform illumination and lacking an one-to-one correspondences from RGB to surface normal we propose a continuous neighborhood based lookup table. This entails a neural network representation (Fig. 5-5) that takes as input the window around the RGB pixel you want to predict the surface normal for, and the coordinate of the RGB pixel. The pixel location allows us to address the non-uniform illumination, and the window allows us to bring in contextual information around the pixel to use more information that hopefully brings an one-to-one correspondence.

5.4.3 Results

To illustrate that a window-based representation improves the quality of the prediction of the surface normal, we benchmark our method with various window sizes. We also show how the pixel location significantly improves our estimates.

First we show that the pixel location significantly helps. We train two models each with a window size of one, but one model is trained without the pixel location. The reported error is the MSE of surface normals estimates on our training set (Table

5.2).

w/o Pixel Location	w/ Pixel Location
0.053	0.018

Table 5.2: Shows the MSE of including or not including the pixel location when estimating surface normal orientation from RGB pixel value.

Now we examine the affect of having a larger window size w on our MSE of the surface normals estimates on our training set (Table 5.3).

w=1	w=3	w=5	w=7	w=9
0.018	0.011	0.009	0.007	0.006

Table 5.3: Shows the influence of aggregating neighborhood information while estimating surface normal orientation from RGB pixel value. In particular the affect of having different window sizes.

While a window size of 9 gives the best performance, in order to avoid adding semantic information, we use a window size of 5 in future use. Future work can include more than a spherical probe to prevent overfitting on spherical data. The resolution of the image is 640x480, so the affect of using a sphere only during training and a window size of 9 has not been quantitatively determined and may not matter. In the case of Fig. 5-4, the length of the ball's major axis in image space is around 50 pixels. It is also worth noting that while interacting with everyday objects like a bottle, the contact patches produced are much larger than the calibration probe used in this experiment. Therefore, in future work using a larger probe to quantify if this method works well for data with low spatial frequency may be necessary.

5.5 3D Reconstruction

Now that the sensor has been calibrated, we need to determine if 3D reconstruction is suitable for real-time feedback in robotic applications. In our prior work [23], we were only able to perform 3D reconstruction on a single sensor at 40 Hz with the

sensors running at 320x240. Since we have four fingers this is not suitable, so we created a GPU accelerated implementation our method where we solve the Poisson equation:

$$\nabla^2 f = g$$

where,

$$g = \frac{\partial}{\partial x}(\frac{\partial f}{\partial y}) + \frac{\partial}{\partial y}(\frac{\partial f}{\partial y})$$

with the Fast Poisson solver with discrete sine transform (DST). After implementing the Fast Poisson solver on a GPU (Nvidia Titan Xp) we are able to get 3D reconstruction at about 600 Hz at 640 x 480. Which is suitable for having four fingers that can run at a max of 90 Hz on our Allegro Hand platform.

Chapter 6

Controlled Rolling

Figure 6-1: Sample begin and end state of each object during the rolling phase of the experiment in the controlled rolling task.

To validate the sensors and the sensor geometry we will perform controlled rolling of a set of unknown objects. In order to do controlled rolling of an unknown object, we propose a tracking method along with a reactive controller to deal with uncertainty in the geometry and dynamics of the object. We assume that no slip will occur throughout execution of the trajectory and chose an action according to the changes in geometry of the sensing surface.

6.1 Method

6.1.1 Tracking Contact Area

To track the contact area we use Iterative Closest Point (ICP) [3]. On each time-step we calculate the current contact area and calculate its convex hull. We get the points from the previous contact area that lie within the convex hull of the current contact area and perform ICP to get the change in the contact area. We only perform ICP with the points within the convex hull since on each time step while rolling we lose contact with areas of the previous contact area.

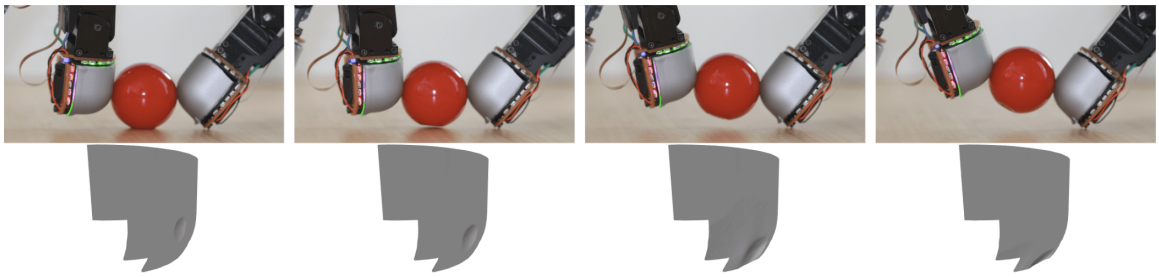

Figure 6-2: **Top from left to right** Sequence of the experiment being performed on the plastic sphere during the rolling stage. **Bottom from left to right** The corresponding point clouds showing the evolution of the point cloud as the object rolls.

6.1.2 Controller

We use a hybrid velocity/force controller [16]. This allows the finger to perform a compliant motion where the finger moves up in task space while maintaining a consistent force normal to the the contact patch, resulting in a rolling motion. The maximum displacement of the contact surface is used as a proxy for force in the controller.

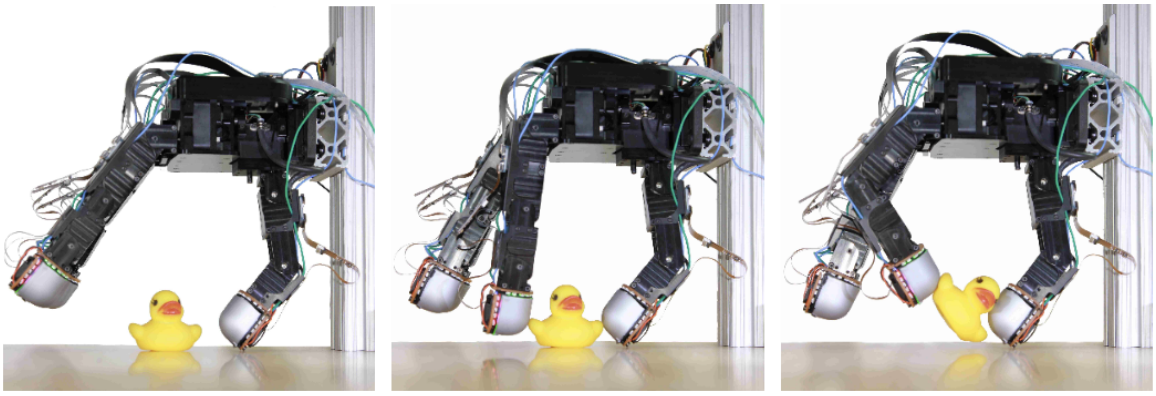

Figure 6-3: Illustration showing the experimental set-up. **From left to right** The placement of the object, when the object makes contact with both fingers, the end state of the object when the object reaches the desired position.

6.2 Results

Experimental Setup

To validate our sensor's capabilities we perform controlled rolling on a set of unknown objects as shown in Fig. 6-1, Fig. 6-3, and Fig. 6-2. The experiment is broken up into three stages. In the first stage we manually place an unknown object in between the index finger and the thumb of the Allegro Hand. While the thumb of the Allegro Hand stays stationary the index finger moves towards the thumb until contact is made. After contact is made, the index finger will continue to move towards the thumb until the desired maximum displacement of the sensing surface is achieved. We then use the controller described in Sec. 6.1.2 to roll the object until contact is made in a desired region of the finger as shown in Fig. 6-2. A trial is considered a success if the finger is able to roll the object until contact is made within the target contact region. If the object falls out of the grasp at any stage in the trial, overshoots the desired contact region, or does not reach the contact area within 3 seconds after the rolling stage of the trial begins the trial is considered a failure. We perform 10 trials for each object.

Experimental Results

As illustrated in table (add table here) our method and sensor were able to successfully perform 99 out of 100 controlled rolls into the desired contact region despite being presented a diverse set of objects with varying smoothness, hardness, and geometries. We show a sample of each object being rolled in Fig. 6-1, Fig. 6-3, and Fig. 6-2. Looking at some of the objects, in particular the roll of tape and yo-yo, the object was successfully rolled to the desired position but exhibits unwanted rotational slip. The object that is the source of the unsuccessful trial was the golf ball. During the execution of rolling the object rolled out of the fingers. This might be a result of the dimples reducing the total contact area made with the sensing surface. Fig. 6-3 serves as visual verification of the 3D reconstruction.

Chapter 7

Cap Removal

To illustrate multi-finger coordination need for dexterous manipulation, we perform a cap removal task (Fig. 7-1). Typically performed with an object fixed on a surface, and performed with a parallel gripper which is performed slowly, we show the sensitivity of the sensors to perform it on a smooth slippery surface, and that the dexterity allows it to be quickly performed in less than 11 seconds.

7.1 Experimental Setup

At each trial the object is randomly placed within the work space of the hand. Due to the limited strength of the hand and only having one hand, the cap is loosened, so that the robot is able to unscrew. The task is performed on a laminated sheet and with a smooth glass jar, so that if too much force is applied to the lid it will slide out of the reach of the manipulator. The trial is considered successful if the robot is able to remove the lid.

7.2 Method

In order to perform this task the robot must localize the jar, bring it into reach of the fingers performing the unscrewing, unscrew the jar while minimizing forces, and then finally knowing when the lid is unscrewed.

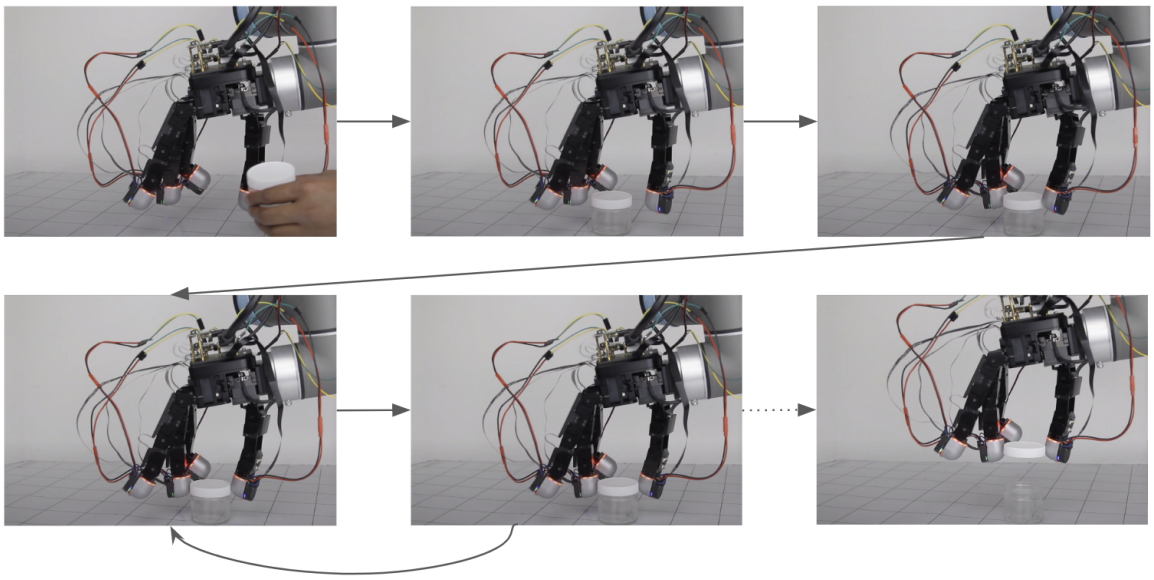

Figure 7-1: Shows the process of cap removal. **From left to right and top to bottom** An object is randomly placed by a human in the workspace of the manipulator. The robot then uses tactile search to localize the jar. Once reaching the ideal location of the jar the robot sweeps it into a position more suitable for unscrewing. The robot then continuously unscrews the jar. Once it detects the jar lid is no longer moving up it removes the lid. The procedure takes less than 11 seconds.

7.2.1 Localization

Once the jar is placed within the work space of the manipulator, the robot must determine its pose. It begins with the robot searching for the jar lid by tapping around. Once the lid is found the robot continues tapping and moving laterally towards the center of the jar. Due to the curvature of the finger and the jar, if the finger is located at the side of the jar it is off the center of the finger which you can observe via the sensor. So once the jar is seen in the center of the finger we know it is in the ideal location for repositioning the jar.

7.2.2 Relocation

For relocation we perform an open-loop sweep towards the center of the the manipulator. The motion of the sweep is limited so feedback wasn't deemed necessary during this stage. Plus during unscrewing, feedback allows the robot to robustly unscrew

the cap despite uncertainty in the initial unscrew pose.

7.2.3 Unscrewing

Similar to the rolling task we employ a hybrid velocity/force controller [16]. So we move tangentially to the lid of the jar, while maintaining a constant normal force. Since the fingers do not estimate force, we use maximum deformation depth as a proxy for force. The motion of the unscrewing is stopped when both fingers no longer detect contact. It then repositions the fingers without collision to the start pose of the unscrewing motion. This is done until it is determined that the lid is unscrewed.

7.2.4 Determining if Unscrewed

On each unscrewing motion and replacement of the fingers to the initial position of the screwing motion, it is assumed that the fingers make contact at the same height. Therefore, at each step in the unscrewing you can determine the change in location in the upwards direction of the lid to see if you still need to unscrew. If the Z-position of the lid did not change then it is assumed that the lid is unscrewed. This is not robust, and ideally, the sensors would be equipped with dots to track tangential forces to see if there is still resistance to unscrewing.

7.3 Results

After performing 16 trials, the robot was able to successfully perform the task 13 times. Causes for failure was caused by either the robot falsely determining that the lid is unscrewed, or the jar was placed to too far from the center of the hand, so while sweeping it the object would slide out of reach.

Chapter 8

Conclusion

In this work we introduced a novel non-planar soft high-resolution tactile sensor. This is achieved by implementing a novel illumination scheme inspired by optical fibers, in which light is constrained along a curved path via an epoxy-air interface, which causes total internal reflection, in opposition to a semi-specular membrane. Once an object is pressed into the membrane, the images from a fisheye camera are used to estimate the surface normals along the deformed surface from changes in surface reflection. These estimated surface normals can finally be used to reconstruct the deformed membrane's geometry via Poisson's equation. This design introduces three main challenges, extreme optical distortion due to the fisheye lens, non-uniform illumination due to partial scattering from the semi-specular membrane upon each reflection, and the bidirectional reflectance distribution function (BRDF) of the semi-specular pigment making mapping an RGB value to surface normal direction difficult. To overcome these issues we introduce a new calibration procedure, as well as methods for learning the fisheye projection model and the RGB-to-surface normal mapping. Of particular interest is the method for learning the RGB-to-surface normal mapping, in which we introduce a neighborhood and location based approach that significantly outperforms methods used on previous GelSight sensors. We finally show the real-world applicability of the sensors by performing two robotic task, the first is controlled rolling of an object placed on a support surface, and the second is lid opening of a randomly placed object on a slippery surface. These task demonstrate the importance

of the shape of the sensor, and how feedback from the sensor can be successfully used in performing localization and the regulation of forces for delicate tasks.

8.1 Future Work

8.1.1 Hardware

While this work introduces a means to produce sensors suitable for dexterous manipulation, due to retrofitting an existing dexterous hand, their functionality is only suited for task that only require the use of fingertips. However, many of the everyday task that humans perform also involve the other segments of the fingers, as well as the palm. Therefore, future work involves engineering a fully sensorized hand. There are two options to accomplish this, a soft hand with a continuous sensing surface along the lines of [28], or a rigid hand with segmented sensing surfaces. While prior work explored articulated continuous sensing surfaces and their benefits, it is still challenging to interpret the tactile image due to wrinkling and deformation. These challenges can be ignored by having segmented sensing surfaces, however due to the size of the camera and the lack of visual features to extract proprioceptive data as in [28], one would have to create a very compact yet stiff mechanical structure.

8.1.2 Software

In terms of software, when it comes to dexterous manipulation, deep reinforcement learning (DRL) is the dominant paradigm. However, the success of DRL for robotics relies heavily on computationally efficient simulation environments [17]. This is problematic for soft tactile sensors since simulating soft-bodies and soft contact dynamics is computationally expensive. Therefore, future work focuses mostly on overcoming this bottleneck. This could be approached in a few ways. First, graph neural networks (GNN) have been used to emulate finite element method (FEM), the method commonly used to simulate soft-body dynamics, multiple orders of magnitude faster than its reference simulation [21]. Learning generalizable contact dynamics of the sensor

with a GNN and using it as a simulation for DRL is a promising direction. Otherwise, one could hand craft a set of elastomer independent features. For instance, the direction of sheer as well as the detection of incipient slip are elastomer independent for quasi-static task. Therefore, estimating these features in current robotic simulators may also be a promising direction.

Bibliography

[1] M. Bauza, O. Canal, and A. Rodriguez. Tactile mapping and localization from high-resolution tactile imprints. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3811–3817, 2019.

[2] Roberto Calandra, Andrew Owens, Dinesh Jayaraman, Justin Lin, Wenzhen Yuan, Jitendra Malik, Edward H Adelson, and Sergey Levine. More than a feeling: Learning to grasp and regrasp using vision and touch. *IEEE Robotics and Automation Letters*, 3(4):3300–3307, 2018.

[3] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, pages 2724–2729 vol.3, April 1991.

[4] Ravinder S Dahiya, Giorgio Metta, Maurizio Valle, and Giulio Sandini. Tactile sensing—from humans to humanoids. *IEEE transactions on robotics*, 26(1):1–20, 2009.

[5] R. Deimel and O. Brock. A compliant hand based on a novel pneumatic actuator. In *2013 IEEE International Conference on Robotics and Automation*, pages 2047–2053, 2013.

[6] Elliott Donlon, Siyuan Dong, Melody Liu, Jianhua Li, Edward Adelson, and Alberto Rodriguez. Gelslim: A high-resolution, compact, robust, and calibrated tactile-sensing finger. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1927–1934. IEEE, 2018.

[7] C Brown Duane. Close-range camera calibration. *Photogramm. Eng*, 37(8):855–866, 1971.

[8] S. Dong F. Hogan, J. Ballester and A. Rodriguez. Tactile dexterity: Manipulation primitives with tactile feedback. In *ICRA*, 2020.

[9] Francois R. Hogan, Michael Jenkin, Sahand Rezaei-Shoshtari, Yogesh Girdhar, David Meger, and Gregory 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 (WACV)*, pages 1218–1227, January 2021.

[10] Roland Johansson. The effects of anesthesia on motor skills - youtube.

[11] Roland S Johansson and J Randall Flanagan. Coding and use of tactile signals from the fingertips in object manipulation tasks. *Nature Reviews Neuroscience*, 10(5):345, 2009.

[12] Naveen Kuppuswamy, Alex Alspach, Avinash Uttamchandani, Sam Creasey, Takuya Ikeda, and Russ Tedrake. Soft-bubble grippers for robust and perceptive manipulation, 2020.

[13] Mike Lambeta, Po-Wei Chou, Stephen Tian, Brian Yang, Benjamin Maloon, Victoria Rose Most, Dave Stroud, Raymond Santos, Ahmad Byagowi, Gregg Kammerer, and 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*, 5(3):3838–3845, Jul 2020.

[14] Nathan F Lepora, Kirsty Aquilina, and Luke Cramphorn. Exploratory tactile servoing with active touch. *IEEE Robotics and Automation Letters*, 2(2):1156–1163, 2017.

[15] Nathan F Lepora and Benjamin Ward-Cherrier. Superresolution with an optical tactile sensor. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2686–2691. IEEE, 2015.

[16] M. T. Mason. Compliance and force control for computer controlled manipulators. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(6):418–432, June 1981.

[17] OpenAI, Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, Jonas Schneider, Nikolas Tezak, Jerry Tworek, Peter Welinder, Lilian Weng, Qiming Yuan, Wojciech Zaremba, and Lei Zhang. Solving rubik's cube with a robot hand. *CoRR*, abs/1910.07113, 2019.

[18] A. Padmanabha, F. Ebert, S. Tian, R. Calandra, C. Finn, and S. Levine. Omnitact: A multi-directional high-resolution touch sensor. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 618–624, 2020.

[19] Radhen Patel, Rebecca Cox, and Nikolaus Correll. Integrated proximity, contact and force sensing using elastomer-embedded commodity proximity sensors. *Autonomous Robots*, 42(7):1443–1458, 2018.

[20] N. Pestell, L. Cramphorn, F. Papadopoulos, and N. F. Lepora. A sense of touch for the shadow modular grasper. *IEEE Robotics and Automation Letters*, 4(2):2220–2226, April 2019.

[21] Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W. Battaglia. Learning mesh-based simulation with graph networks. *CoRR*, abs/2010.03409, 2020.

[22] Branden Romero Radhen Patel, Rui Ouyang and Edward Adelson. Digger finger gelsight tactile sensor for object identification inside granular media. In *ISER*, 2021.

[23] Branden Romero, Filipe Veiga, and Edward Adelson. Soft, round, high resolution tactile fingertip sensors for dexterous robotic manipulation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4796–4802. IEEE, 2020.

[24] Francisco Romero-Ramirez, Rafael Muñoz-Salinas, and Rafael Medina-Carnicer. Speeded up detection of squared fiducial markers. *Image and Vision Computing*, 76, 06 2018.

[25] D. Romeres S. Kim D. Nikovski S. Dong, D. Jha and A. Rodriguez. Tactile-rl for insertion: Generalization to objects of unknown geometry. In *ICRA*, 2021.

[26] Carmelo Sferrazza and Raffaello D'Andrea. Design, motivation and evaluation of a full-resolution optical tactile sensor. *Sensors*, 19(4), 2019.

[27] Carmelo Sferrazza, Adam Wahlsten, Camill Trueeb, and Raffaello D'Andrea. Ground truth force distribution for learning-based tactile sensing: a finite element approach. *CoRR*, abs/1909.04000, 2019.

[28] Yu She, Sandra Q. Liu, Peiyu Yu, and Edward Adelson. Exoskeleton-covered soft finger with vision-based proprioception and tactile sensing. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10075–10081, 2020.

[29] Yu She, Shaoxiong Wang, Siyuan Dong, Neha Sunil, Alberto Rodriguez, and Edward Adelson. Cable manipulation with a tactile-reactive gripper, 2020.

[30] Edward J. Smith, Roberto Calandra, Adriana Romero, Georgia Gkioxari, David Meger, Jitendra Malik, and Michal Drozdzal. 3d shape reconstruction from vision and touch, 2020.

[31] Chen Wang, Shaoxiong Wang, Branden Romero, Filipe Veiga, and Edward Adelson. Swingbot: Learning physical features from in-hand tactile exploration for dynamic swing-up manipulation, 2021.

[32] Shaoxiong Wang, Jiajun Wu, Xingyuan Sun, Wenzhen Yuan, William T Freeman, Joshua B Tenenbaum, and Edward H Adelson. 3d shape perception from monocular vision, touch, and shape priors. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1606–1613. IEEE, 2018.

[33] Akihiko Yamaguchi and Christopher G Atkeson. Combining finger vision and optical tactile sensing: Reducing and handling errors while cutting vegetables. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 1045–1051. IEEE, 2016.

[34] W. Yuan, C. Zhu, A. Owens, M. A. Srinivasan, and E. H. Adelson. Shape-independent hardness estimation using deep learning and a gelsight tactile sensor. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 951–958, 2017.

[35] Wenzhen Yuan, Siyuan Dong, and Edward Adelson. Gelsight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12):2762, 2017.

[36] Wenzhen Yuan, Rui Li, Mandayam A Srinivasan, and Edward H Adelson. Measurement of shear and slip with a gelsight tactile sensor. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 304–311. IEEE, 2015.

[37] Wenzhen Yuan, Yuchen Mo, Shaoxiong Wang, and Edward H Adelson. Active clothing material perception using tactile sensing and deep learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8. IEEE, 2018.