Simulation-guided Design for Vision-based Tactile Sensing on a Soft Robot Gripper

Yichen Li

CMU-RI-TR-23-20

May 1, 2023

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Wenzhen Yuan, *chair*
Matthew O'Toole
Arpit Agarwal

Submitted in partial fulfillment of the requirements for the degree of Master in Robotics.

To the stormy sea I learned to sail.

Abstract

Soft pneumatic robot manipulators have garnered widespread interest due to their compliance and flexibility, which enable soft, non-destructive grasping and strong adaptability to complex working environments. Tactile sensing is crucial for these manipulators to provide real-time contact information for control and manipulation, but most tactile sensors are either inflexible, low-resolution, expensive, or hard to manufacture.

Vision-based tactile sensing has been used in soft pneumatic robot manipulators to achieve a high spatial resolution of contact information. However, designing a high-quality vision-based tactile sensor for soft pneumatic robot manipulators is challenging for two reasons. Achieving steady high-fidelity sensing signals on a deformable robot is non-trivial, and manufacturing time and expense make design iteration in the real world unpractical.

In this thesis, we present a physics-based optical simulation pipeline to guide the design of vision-based tactile sensing on a soft robot finger. Our simulation pipeline enables faster iteration cycles and automatic optimization of design parameters, eliminating the need to manufacture the entire robot finger for each design iteration. The simulation utilizes physics-based rendering (PBR) with highly accurate optical modeling of the robot finger to ensure that performance improvements in tactile sensing in the simulation are transferable to the real-world robot finger. We introduce a fast numerical metric to test tactile sensing performance at different robot states and contact locations to evaluate designs. During the optimization, we perform a grid search on possible combinations of light color choices for optical fibers based on defined metrics. Then, we apply the covariance matrix adaptation evolution strategy (CMA-ES) as a numerical optimization method to iteratively fine-tune the directions of the optical fibers. We compare the tactile sensing performance of the optimized design with the baseline design and demonstrate an improvement.

This thesis makes the following contributions: a) a ready-to-use simulation pipeline for any vision-based tactile sensor; b) providing a new design paradigm that uses optical simulation as a testing and optimization platform; c) a new metric for evaluating the sensing performance of vision-based tactile sensors.

Acknowledgments

First of all, I would like to thank Prof. Wenzhen Yuan for providing me with guidance and insightful discussion throughout my entire master's program. She patiently introduced me to tactile sensing, a whole new area that I have little experience with, and supported me throughout the learning process. She always challenged me to think deeply and push the limit and I would never accomplish all of these without her encouragement.

I am also grateful to Prof. Matthew O'Toole and Arpit Agarwal. I learned a lot about computer graphics and physics-based rendering which becomes a crucial part of this thesis. They are always there, ready to help and provide feedback.

Lastly, I am so grateful to work with all the Wenzhen's lab members. A huge shout out to Uksang Yoo, Joe Huang, Ruihan Gao, Achu Wilson, Yufan Zhang, Xiaofeng Guo, Helen Jiang, Arkadeep Chaudhury, and Zilin Si. They are the most friendly and supportive collaborators I have ever met. I also want to thank my parents for providing all the emotional support along this journey.

Contents

List of Figures

Chapter 1

Introduction

Soft robots are an emerging class of robots made out of soft and compliant materials [1]. The deformable nature of the soft robots enables them to have a higher degree of freedom, better flexibility for certain tasks, and stronger adaption to the working environment. Thus, it has been widely used in agriculture and human-robot interaction where compliant and non-destructive interactions are required [2] [3].

Soft robots are different from rigid body robots in that they could easily warp around objects and comply with the environment, making contact detection and localization important for accurately determining their shape and position. Besides that, the deformable surface of soft robots could directly provide detailed contact surfaces' local geometry features, which is almost impossible for rigid robots. Take a soft robot gripper for example, we could use tactile information to decide whether the gripper has already touched the object and where the contact happens [4], local deformation to estimate the grasping force and detect slipping [5] [6], and local geometry feature to classify the contact object or estimate the shape of the contact object[7]. Therefore, incorporating tactile sensing is advantageous for soft robots.

1.1 Related Works and Motivation

Tactile sensing provides rich information about the robot, but enabling tactile sensing on soft robot grippers is challenging. Soft and flexible tactile sensors are necessary due to the challenges posed by surface deformation. The use of rigid components

in onboard tactile sensors can largely constrain the actuation of the robot, limiting its ability to bend or extend freely. Thus, soft and flexible tactile sensors are widely applied for soft robot tactile sensing. Many works proposed the use of flexible tactile sensors for tactile sensing on soft robotic grippers [8] [9]. Truby et al. [8] proposed embedded ionogel sensors into soft robotics fingers for tactile sensing [8]. These sensors are made from conductive ionogel ink embedded into molded network inside the robot's surface. When the robot inflates or contacts objects, the local geometry change will affect the resistance of the ionogel sensor, and this change can be used to read out tactile information. Because the tactile sensing resolution is directly proportional to the number of the ionogel sensor embedded, ionogel sensors are hard to provide high-resolution tactile information limited by the difficulty of molding complicated ionogel networks into the robot surface. Pannen et al. [9] also proposed a flexible tactile sensor for soft robots that uses low-cost piezoresistive fabric for tactile sensing. The sensor consisted of a piezoresistive fabric layer sandwiched by two electrode layers and it covered the entire soft robot surface. Similar to ionogel sensor, the piezoresistive sensor also relied on the resistance change of the piezoresistive layer while contacting to gain tactile information. It achieves a relatively high spatial resolution of 4.5mm at a relatively low cost in manufacturing, however, the fabric layer and flexible PCB electrode layers were not flexible enough to comply with the drastic deformation of the soft robot.

On the other hand, vision-based tactile sensors could provide extremely high sensing resolution at a low cost and do not require installing any device on the sensing surface of the robot. Yu et al. [10] demonstrated a soft robot finger with vision-based tactile sensing. To ensure the best coverage and lighting conditions, the soft robot finger is divided into several sections connected by rigid joints where each section is fully covered by a camera and LED light source on the side. Sensing surface shape could be inferred by the joint angles and deformation on the sensing surface could be extracted from camera images. However, this approach has two major drawbacks: the robot finger is constrained by joints and the single-color LED does not provide enough information for inferring surface normal. Another work on vision-based tactile sensors with improvement on both drawbacks was by Liu et al.[11]. They used reference points on the surface to estimate the sensing surface shape so that the robot could freely deform without any joint constraints, and acrylic paint on the side wall will

reflect different colors of lights on the sensing surface to better visualize the surface deformation. However, one limitation of the design is that the only active light source is placed at the base of the soft robot. When the robot deforms, the sensing area far from the light source will receive much less illumination and downgrade the sensing performance.

The proposed tactile sensors for soft robots lack a systematic design and sensor performance evaluation procedure. One of the possible reasons is that design iteration usually takes a long time due to the manufacturing process. The molding and curing process for the soft robot body, assembling parts, and calibrating the sensor often takes days to complete. At the same time, there doesn't exist an efficient sensor design evaluation procedure for tactile sensors. Therefore, we would like to address these problems in this thesis.

1.2 Thesis Overview and Contribution

In this thesis, we present a new simulation pipeline for rapidly iterating on vision-based tactile sensing design on a soft robot gripper. It consists of:

- A background on vision-based tactile sensing and highlight the challenges in the design of our soft robot gripper. In Chapter 2, we will introduce the working principle of vision-based tactile sensing and develop an understanding of what contributes a good tactile sensing from theory. Then, we explain the design challenges in the context of the physical design of our soft robot finger and state our design goal.
- An overview of the optical simulation tool. In Chapter 3, we will introduce an optical simulation tool that uses physics-based rendering to generate realistic simulation of arbitrary robot configuration. An overview of physics-based rendering will be provided to illustrate its unique advantage in providing high-fidelity simulation results that transfer to real-world prototypes. We will discuss the specific rendering algorithm choice essential to our robot setup and how we model the optical properties of each component of our robot.
- An overview of a two-stage automated design optimization process. In Chapter 4, we will formulate our design goal and introduce our optimization process for

vision-based tactile sensing design on soft robots. A simple but theory-grounded metric for evaluating sensing performance is proposed and a two-stage design optimization will explore both lighting color choice and lighting direction to maximize the performance of our design in multiple robot configurations.

Our main contributions are:

- A new simulation-guided design workflow that largely reduces the turnaround time by eliminating the manufacturing time.
- A new metric for evaluating the sensing performance of vision-based tactile sensors, which is grounded in photometric theory and quick to evaluate.
- A robust automated design optimization process for iterating on sensor designs.

Chapter 2

Design Problem Formulation

In this Chapter, we present the challenges of designing vision-based tactile sensors on soft robot grippers and formulate the challenges into a design optimization problem. We begin by providing background knowledge on vision-based tactile sensing and describing the physical design of our soft robot gripper. Then, we offer insights into how the robot gripper's shape and deformation can affect the sensing quality and use these insights to formulate a design optimization problem.

2.1 Theory of Vision-based Tactile Sensing

The idea of vision-based tactile sensing stems from photometric stereo, a technique for estimating surface normal by using a series of images taken with different directions of incident illumination [12]. Suppose that intensity of a pixel at (x, y) is $I(x, y)$ and the surface normal at the same pixel is (p, q), $I(x, y) = R(p, q)$ by some function R that captures the surface's bidirectional reflectance distribution function (BRDF). The goal is to learn the spatial mapping of surface normals, denoted as $(p, q) = f(x, y)$. Since the function R is non-linear, it requires three images by illumination of different directions to solve the function f. Alternatively, one image can be taken with three colored lights incident from different directions on the sensing surface. At its core, the problem of photometric stereo is to find a normal estimation function f to estimate surface normal

$$n = f(I)$$

, where I is a $x \times y \times 3$ RGB image and n is a $x \times y \times 2$ normal map, x, y is the number of rows and columns of the image. For example, if the surface material obeys the Lambertian model, which $I_i = \sigma |w_i \cdot n|$ assuming unit incoming illumination irradiance, we can directly solve for the f. Construct a 3×3 matrix $N = \begin{bmatrix} w_r \\ w_g \\ w_b \end{bmatrix}$ where w_i is incident direction for $\{red, green, blue\}$ colored light. Then, $\sigma = |N^{-1}I|$ and $n = \frac{1}{\sigma} N^{-1} I$ if N is invertible.

One problem in determining the normal estimation function f is its high correlation with the BRDF of the object's material. If the BRDF is highly non-linear or even non-continuous, then f is extremely hard to estimate. To address this issue, vision-based tactile sensors, take Gelsight for example, closely fit the surface of the object with a deformable elastomer layer coated with Lambertian-like material inside facing the camera and light sources. This effectively replaces the surface material property with Lambertian-like material so that normal estimation results are independent of the type of contacting objects. Additionally, vision-based tactile sensors fix the illumination condition by blocking the environment light and providing fixed colored LEDs with desired angles, making the normal estimation function is consistent regardless of the working environment.

Once the BRDF of the surface has been fixed, then the only remaining factor for determining the normal estimation function is the illumination condition, more specifically incident lighting radiance and directions. Assume the surface is Lambertian and only consider $\omega_i \cdot n > 0$, we have

$$\begin{aligned} I &= \sum_i \int_\Omega \sigma L_i(\omega_i) \omega_i \cdot n d\omega_i \\ &= \sum_i \int_\Omega \sigma L_i(\omega_i) \otimes \omega_i d\omega_i n \\ &= \mathcal{R}(n) n, \end{aligned} \tag{2.1}$$

where light source i has radiance $L_i(\omega_i)$ at direction ω_i. The reason that $\mathcal{R}(n)$ depends on normal vector n is because the range of valid incident light directions Ω is determined by n. Let r_i be ith row of $\mathcal{R}(n)$. Here, we make an assumption that RGB color vectors reside in a vector space $[0, 255]^3$ and color channels are orthogonal

to each other. Let's define ensemble irradiance matrix as $\mathcal{K} = diag(\begin{bmatrix} ||r_0|| \\ \cdots \\ ||r_m|| \end{bmatrix})$, and ensemble direction matrix as $\mathcal{N} = \begin{bmatrix} \frac{r_0}{||r_0||} \\ \cdots \\ \frac{r_m}{||r_m||} \end{bmatrix}$ for a m row matrix $\mathcal{R}$. The physical meaning of r_i is the sum of all vectors v_j, where $\frac{v_j}{||v_j||} = \omega_j$, $||v_j||$ is the ith RGB color vector.

This formalism provides an interesting perspective on vision-based tactile sensor design by analyzing both the ensemble irradiance matrix and the ensemble direction matrix. In theory, we only need the rank of $\mathcal{N} \geq 2$ to be able to uniquely map every RGB color vector to a normal vector because normal vectors only have two degrees of freedom. Assuming we only consider incident light direction $\omega_i \cdot n > 0$, two lights of different colors from distinct directions will satisfy the requirement. However, considering how limited the range of incident light direction will satisfy the assumption, it is always preferable to have all locations on the sensing surface reachable by all three colored lights.

However, having more than two distinct light directions of each color may result in colinearity between the rows of $\mathcal{N}$, causing a loss in rank. Additionally, the image noise and oversimplified material property assumption can also add uncertainty to the color measurement, making it crucial to design a system providing one-to-one mapping as well as error tolerance. Large angles between row vectors of $\mathcal{N}$ can reduce the normal estimation error brought by the uncertainty in color measurement. Thus, a good design should ensure each normal vector n_i has at least lights of two colors from distinct ensemble directions $\hat{w}_1 \cdot n_i > 0, \hat{w}_2 \cdot n_i > 0$ where $\hat{w}_i = \frac{r_i}{||r_i||}$, and the angle between ensemble directions should be close to orthogonal.

In fact, the matrix $\mathcal{K}$ also plays an important role in controlling the noise and error of color measurement. Assume our normal estimation by $\tilde{n} = f(\tilde{I})$, where $\tilde{I} = I + \epsilon$. Matrix $\mathcal{K}$ controls the magnitude of mapping f where large entry $\mathcal{K}$ means it requires a large change in color ΔI to have the same change in normal vector Δn. Thus, large entries in $\mathcal{K}$ are in favor of noise reduction. This implies that we should place the same colored light source close together to enhance the irradiance in the dominant direction and any light of the same color coming from the opposite direction will

reduce the robustness of our system.

Thus, we summarize our design requirement for vision-based sensors from theory:

- For any location on the sensing surface, a good design provides lights of as many colors as possible
- The ensemble light direction of each color at all normal vectors should at least be linearly independent of each other. Orthogonal ensemble light direction of each color gives the best result.
- Good design should have the incident directions of light of the same color be as close as possible to enhance the radiance along the ensemble direction.

2.2 Physical Design of our Soft Robot Gripper

In this section, we will introduce the physical design of the soft robot gripper built in our lab by Uksang Yoo, shown in the schematic Fig 2.1. Our soft robot gripper

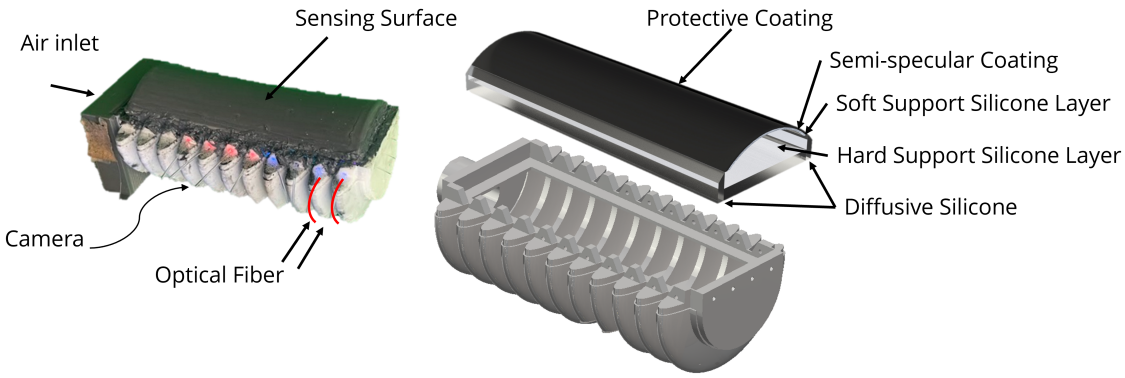


Figure 2.1: Physical design of our soft robot gripper. The main robot parts are shown on the left side and the detailed layer decomposition of the sensing surface is shown on the right side

utilizes pneumatic actuation, where internal air pressure controls the bending of the robot to grasp objects. The robot has two main components: a flexible Thermoplastic Polyurethane (TPU) 3D-printed body with multiple air chambers that can expand, and a flexible, non-expandable silicone sensing surface. When actuated, the robot

only bends towards the sensing surface. To visualize local deformation on the sensing surface, we mount a 160-degree field-of-view camera on the 3D-printed body, facing the sensing surface. We use optical fibers to provide illumination for our sensor, as they are small and flexible enough to not interfere with robot actuation. Specific holes on the edges of the 3D-printed body secure the position and direction of the optical fibers with respect to the sensing surface to prevent changes in light conditions when the robot deforms.

The sensing surface is where local deformation happens and is perceived and it has a novel structure to enhance the sensing quality, shown on the right of Fig 2.1. The top layers consist of a semi-specular coating that reflects light to the camera and a protective coating that blocks outside environment lighting and protects the semi-specular coating. Beneath the top layers, a thin soft support silicone layer allows for large surface deformation, enabling the semi-specular coating layer to closely follow the local geometry of the contacting surface. The bulk of the sensing surface is made up of hard silicone layers, which are hard enough to maintain the shape of the robot while in contact, preventing the un-contacted sensing surface from being pushed inwards by nearby contacts. On both sides of the hard support silicone layer, we attached opaque diffusive silicone layers to distribute the light from the optical fibers more evenly, as light from the optical fibers is highly centralized within a small range of the outgoing direction and large intensity differences along the sensing surface can negatively impact the consistency of tactile sensing.

2.3 Optimization Problem Statement

Designing a good tactile sensor on a soft robot gripper is a challenging task due to the robot's shape and deformable nature. Our soft robot gripper is designed to be long so that it could easily bend and warp around the object, but the large length-width ratio could create an ill illumination condition for vision-based tactile sensors. Take the sensor drawn in Fig 2.2 for example, indentation that happened closer toward the red light source receives much higher red color intensity at deformation than the indentation further away from the red light source. With almost no red light from the left, the same pair of the normal vectors of the same indentation n_1 and n_2 encoded a much similar color (blue and blue) far from the light source than the pair (blue and

pink) closer to the red light source, leading to a worse tactile sensing performance as it is harder to distinguish the two normal vectors. The illumination problem becomes

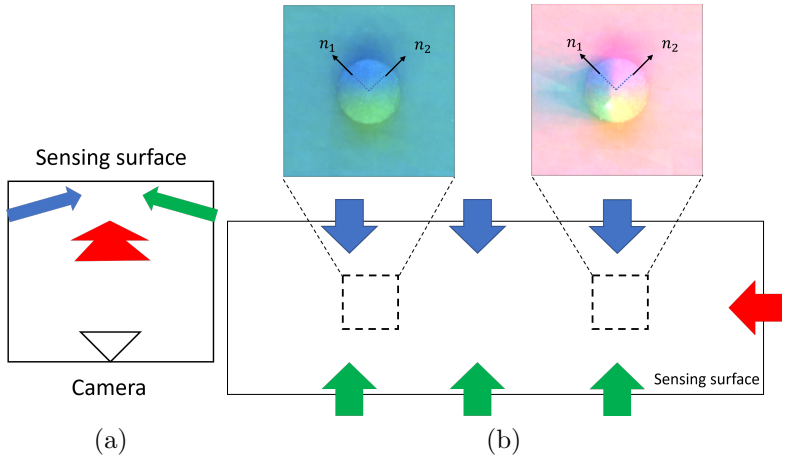


Figure 2.2: Example of how the sensor's shape could affect tactile sensing performance. (a) is the side view of our example tactile sensor. The light source comes from three directions toward the center of the sensing surface and a camera is placed at the bottom of the sensor looking at the sensing surface. (b) shows two image patches of the sensing surface contacted with the same ball indentor at two locations along the red light direction.

worse when the robot gripper bends. Because our light propagation direction does not bend with the robot gripper, the part of the sensing surface that could receive red light from the side will quickly diminish and shift toward the right side of the paradigm, and so is the sensing performance.

For the rest of this thesis, we will focus on addressing the design optimization problem that achieving good tactile sensing performance uniformly across the sensing area under all deformations of the robot finger.

Chapter 3

Optical Simulation Pipeline

In Chapter 2, we introduced vision-based tactile sensing and a potential approach to integrating vision-based tactile sensing on a soft robot gripper. However, the deformable nature and shape of the soft robot gripper pose questions on how to design an effective sensor

3.1 Theory of Physics-based Rendering

Physics-based rendering (PBR) is an image generation technique that models the entire process of light traveling through a scene and reaching the camera sensor according to the laws of physics. Unlike other rendering techniques, PBR can produce highly accurate images that closely match those captured in real-world conditions. This level of accuracy is crucial for simulations where design changes made in the virtual environment should result in corresponding changes in the real world. However, modeling all of the interactions of light with a scene can be challenging and time-consuming due to the complexity of the scene.
Physics-based rendering treats light interaction with objects as the following equation:

$$L_o(x, \omega_o) = L_e(x, \omega_o) + \int_\Omega f(x, \omega_o, \omega_i) L_i(x, \omega_i) \omega_i \cdot n d\omega_i \tag{3.1}$$

, where x is the position where light intersects with the object, ω_o is the outgoing light direction, ω_i is the incoming light direction, Ω is the set of all possible incoming light

direction, n is the surface normal vector of the object, L_e is the emitted light radiance from the object, L_i is the incoming light radiance, L_o is the outgoing light radiance, f_r is a bidirectional scattering distribution function (BSDF) models the proportion of the incoming light radiance contributes to the outgoing radiance. In essence, the equation tells us that outgoing radiance is the sum of emitted light radiance and some portion of the incoming light radiance. Using the same equation, we could substitute incoming light radiance with the outgoing light radiance at another location, shown in the following equation:

$$\begin{aligned} L(p_1 \to p_0) = {} & L_e(p_1 \to p_0) \\ & + \int_A L_e(p_2 \to p_1) f(p_2 \to p_1 \to p_0) dA(p_2) \\ & + \int_A f(p_2 \to p_1 \to p_0) \int_A L_e(p_3 \to p_2) f(p_3 \to p_2 \to p_1) dA(p_3) dA(p_2), \end{aligned} \tag{3.2}$$

where p_i are the locations where light incidents with the scene. With the above equation, we could design a recursive algorithm that computes the radiance of light rays reaching the camera film as a recursive sum of the radiance of each preceding light path. This type of algorithm is also known as the ray tracing algorithm that has been used in most physics-based rendering systems. However, the current version of the algorithm will trace each possible incoming light path at each recursive step and the number of traced paths will increase exponentially because of the curse of dimensionality. Therefore, we use Monte Carlo integration as the approximation sum of all the incoming light radiance in practice. The result of Monte Carlo integration converges quickly to the actual sum. With Monte Carlo integration, we can revise our recursive algorithm as follows:

The final rendering quality will be dependent on the material's optical property, BSDF f, and light path sampling strategy because BSDF will determine the fraction of light that will remain after interacting with the object surface and the sampling strategy will determine the error of the Monte Carlo integration from the expectation value. In section 3.3, we will discuss how we model the optical property of each part of the soft robot gripper and the specific rendering algorithm we used to sample the light path.

Algorithm 1 Ray Tracing Algorithm

```
1: procedure RAY TRACING
2:     depth = d                                              ▷ recursive depth
3:     scene constructed from specs
4:     for camera ray: ((x, y), ω) do
5:         color = RecursiveColor(scene, ω, depth)
6:     end for
7: end procedure
8: procedure RECURSIVECOLOR(scene, ω,depth)
9:     P = scene.intersect(ω)                          ▷ Find where light intersects
10:    if depth == 0 then return Le(P,-ω)
11:    end if
12:    ω′ sampled from BRDF at P
13:    return L ← Le(P,-ω) + BRDF * RecursiveColor(scene, ω′, depth-1) * dot(n,ω′)
   / PDF(ω′)
14: end procedure
```

3.2 Optical Rendering Algorithm

As stated in the previous section, we use Monte Carlo integration to solve the exponential increase in traced light path in the rendering process. Effective sample light path plays an important role in achieving high-quality images with less rendering time. However, our scene is highly complicated with several different optical properties, including diffusive surface reflection, dielectric surface refraction, and volumetric scattering. As shown in figure 3.1, any lights that reflect from the contact surface will follow a similar light path: lights emitted from optical fiber will first be refracted by silicone layer and scattered inside the diffusive silicon layer; After being reflected by the semi-specular contacting surface, lights will undergo another refraction before reaching to the camera sensor. Light paths through multiple dielectric refraction and medium scattering is very hard to solve and exceed the rendering capacity of regular ray tracing algorithm because most light paths generated by randomness are not valid by refraction laws. Thus, we use Primary Sample Space Metropolis Light Transport (PSSMLT) [13] as our rendering algorithm. The algorithm perturbs the existing successful light paths to find the new light paths which have a higher chance to find a successful light path that connects the light sources with the camera.

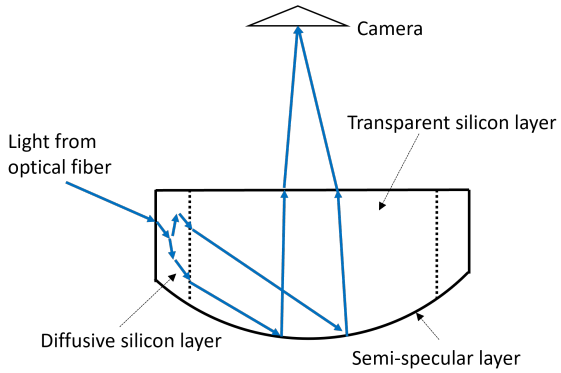
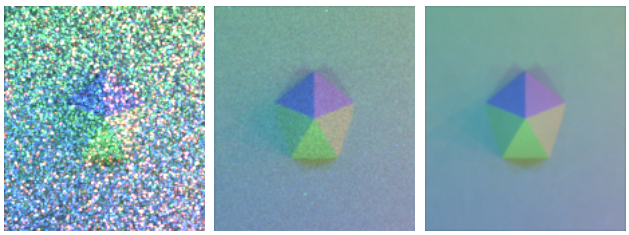


Figure 3.1: Typical light paths from optical fiber to camera.



(a) (b) (c)

Figure 3.2: Rendering results of different rendering algorithms. (a), (b), (c) are rendering results by path tracer algorithm, bidirectional path tracer algorithm, PSSMLT algorithm respectively with 16384 samples per pixel and rendering time for each algorithm are: Path tracing: 3.683 min, BDPT: 5.717 min, PSSMLT: 2.26 min with AMD EPYC 7742 64-Core Processor.

figure 3.2 shows the performance of path tracer, bidirectional path tracer and PSSMLT algorithm implemented in Mitsuba. Path tracer traces light path starting from the camera and uses simple random sampling for path search so many paths found are biased, resulting in a very noisy image. Bidirectional path tracer traces light path starting both camera and light sources so it has a higher chance to find a valid light path, leading to a cleaner but still noisy image than that by the path tracer. PSSMLT performs the best among the three algorithms as it actively searches for valid paths close to existing success light paths.

However, all images directly from MLP-type of algorithms are relative to some unknown scale, so mitsuba's PSSMLT implementation will use another Monte Carlo estimator to determine the scale of the final result [14]. Because the scale determination process is also based on random sampling, we might have scale differences with the same setting. Here, we present rendering results of the same object with different numbers of samples for the scale determination process, which is controlled by "luminanceSamples" in Mitsuba, in figure 3.3. With "luminanceSamples" equal to 1e7, the standard deviation of average pixel intensity of the rendering results converges to a negligible $\approx 0.2\%$.

3.3 Optical Modeling for Soft Robot Gripper

To ensure accurate simulation results, it is necessary to model the optical properties of all components of the soft robot gripper that can interact with light. This includes the optical fibers, diffusive elastomer layer, transparent silicone gel layer, murky diffusive silicone layer, and camera. However, the soft robot body will not be modeled, as we have painted its surface black, which absorbs all light from the inside and blocks all light from the outside environment. In the following paragraphs, we will discuss each part in detail.

3.3.1 Optical Models

Optical fibers are waveguides that could transport electromagnetic waves through them with little to no energy loss[15]. The electromagnetic wave profiles allowed to transport through optical fibers are called guided modes and are determined by the

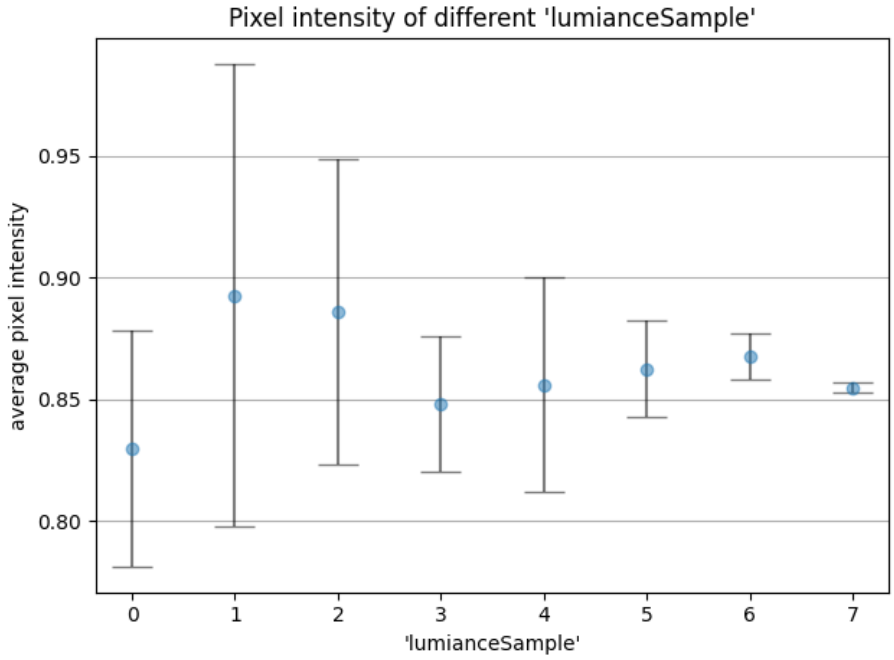


Figure 3.3: Average pixel intensity of the same scene with different "luminanceSamples".

diameter and material of the optical fibers [16]. The optical fibers used in our robot have relatively large diameters ($\approx 0.75mm$ in diameter) which allows a large number of guided modes so that the full energy profile could be transported. Consequently, the distribution of the outgoing radiance of the optical fibers entirely relies on the distribution of radiance of the source LEDs, which is a normal distribution [17]. Due to the small diameters of the optical fibers compared to the scale of our robot, we approximate the end of the optical fiber as a point light source with a specific radiance profile. The radiance profile is a normal distribution centered at direction ω_d with a variance of σ_v and radiance L_{max},

$$L(\omega) = L_{max} \frac{e^{-(\frac{\omega-\omega_d}{2\sigma_v})^2}}{\sigma_v \cdot 2\pi}$$

.

The diffusive silicone layer is made of translucent material, which could be approximated with a homogeneous scattering material. Extensive studies have been

conducted on modeling and measuring this kind of material [18] [19] [20]. In our simulation, we find that the most basic scattering model could actually capture the light scattering effect well enough. The scattering model we use has three parameters: σ_s is the scattering coefficient for controlling the average amount of scattering event; σ_a is the absorption coefficient for controlling the average amount of light absorption; g is the parameter for the Henyey-Greenstein phase function describing the deviation of light directions after each scattering event.

The light models of other materials are straightforward. We use the rough dielectric model to model the transparent silicone layer because the rough dielectric model is suitable for a homogeneous transparent material with a slightly uneven surface. The elastomer layer is modeled by a surface diffusive model in the previous work by our lab[21]. The wide-angle camera can be directly modeled as a perspective camera with a certain field of view and resolution. We neglect the geometry distortion because we will preprocess all the images with the de-warping method for fisheye cameras provided by Opencv library during actual normal prediction.

3.3.2 Calibration

With optical models for each part of the soft robot gripper, we further validate our simulation result with real-world in a fully assembled prototype. The goal of this step is to confirm simulation results are well aligned with the real-world data when all components of our soft robot are involved and robots are in different actuation stages. We replace the soft robot body with a rigid 3D-printed structure in the real prototype setup to ensure the shape of the sensing surface, where all light interactions happen, matches the simulation. This can largely reduce the error coming from the shape estimation of a deformable robot and allow us to focus on calibrating the optical properties of each component. The setup in the real world and simulation is shown in figure 3.4. In the first step, we aim to validate the simulation result with only one optical fiber connected to the light sources. It is to validate the light intensity profile after it comes out of optical fiber and goes through the diffusive silicone material. The imaging result obtained when all optical fibers are connected to the light sources will be a superposition of imaging results obtained with each optical fiber individually, as per the superposition principle of light. Given the symmetric radiance profile of

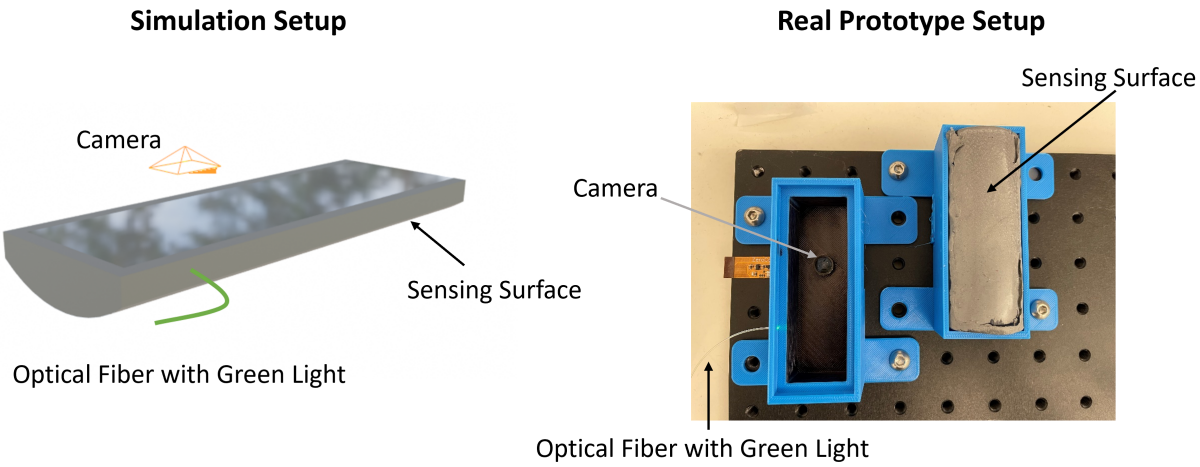


Figure 3.4: Validation setups in both simulation and the real world.

optical fiber and the homogeneous property of the diffusive silicone material and elastomer layer, the image pattern will be symmetric. Thus, It is sufficient to use pixel values along several reference lines symmetrically across the image from both the simulation and real-world to completely describe the image pattern and compare the results. Specifically, the horizontal lines measure the fade-off curve from the center of the image to the edge, while the vertical lines measure the intensity attenuation along the light direction. The results for two robot bending states are shown in figure 3.5.

In general, the curves of simulation and real-world results match. We have seen some mismatches in the simulation result and real-world data. Although the image pattern should be symmetric across the light direction, it does not exactly show in real-world data. The possible reasons are that the distribution of the small reflective coating particle on the elastomer layer is not uniform and the surface of the elastomer layer has bumps that direct light into other directions.

Finally, we compare the image result with all optical fibers connected to light sources in figure 3.6. The overall distribution of the colors in the simulation matches with real-world data, but we also clearly observe that real-world data is not as smooth as the simulation data. It is expected for the same reasons as above in addition to the intensity variance of the light sources that are connected to each optical fiber.

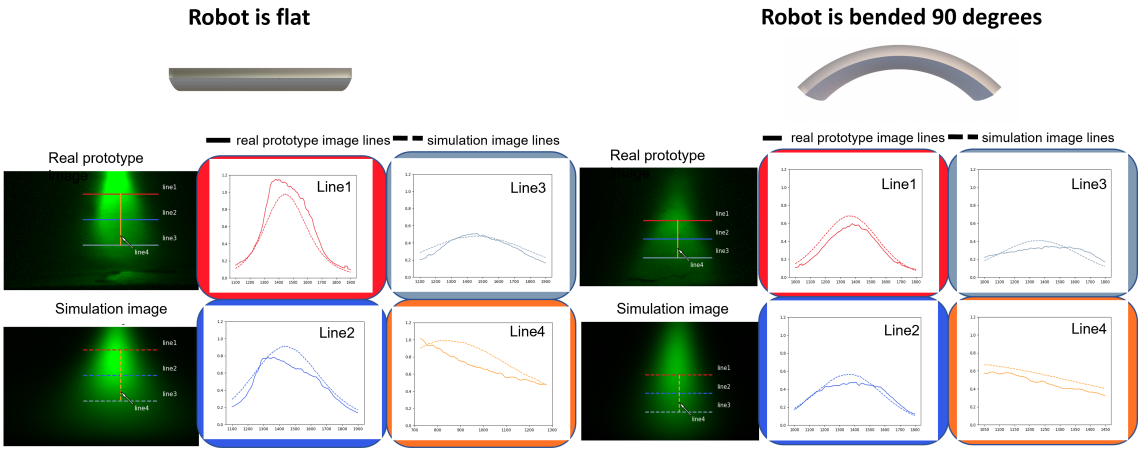


Figure 3.5: Validation results for the whole setup with one optical fiber are connected when the robot is in the flat or rest state and the robot is bent 90 degrees. For each image, we draw four reference lines covering the illuminated area (three horizontal lines capture the intensity changes perpendicular to the light direction; one vertical line captures the intensity changes along the light direction). The intensity along each reference line is plotted on the right image with solid lines representing intensity from real prototype images and dashed lines representing intensity from simulation images

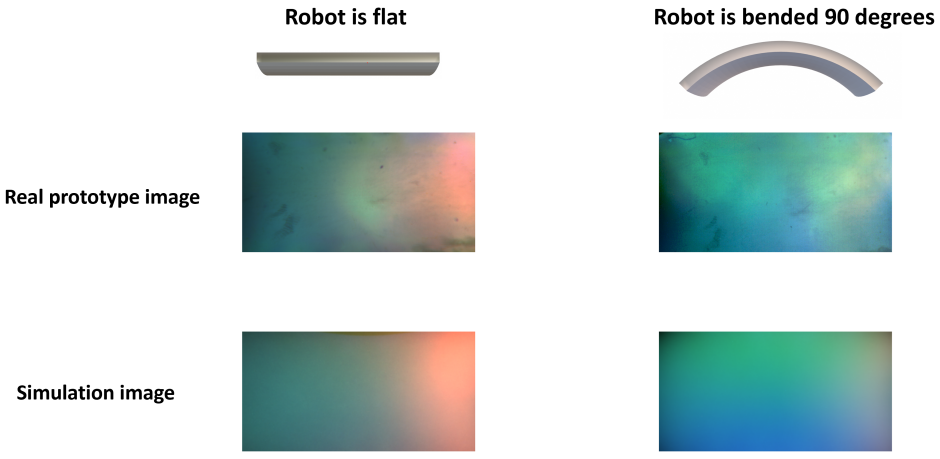


Figure 3.6: Qualitative comparison of images from real prototype and simulation.

Chapter 4

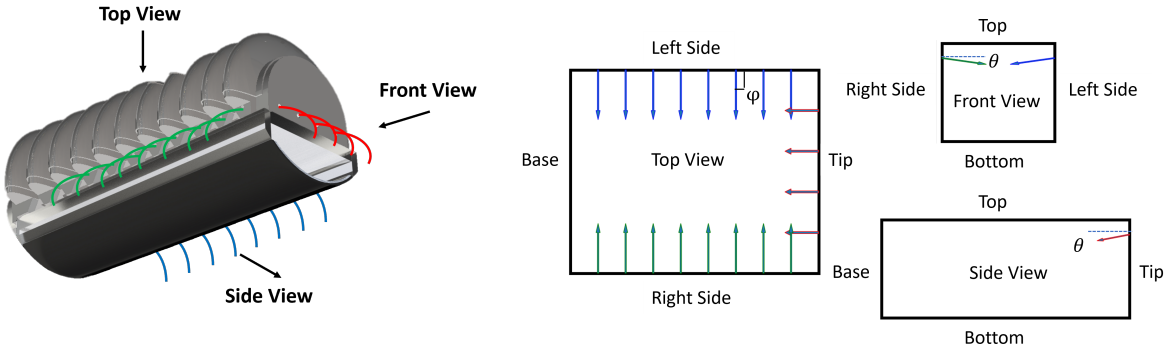
Design Optimization

In the previous chapter, we proposed our simulation pipeline and validated it through optical model calibration and rendering algorithm exploration. In this chapter, we will formulate our design optimization for the vision-based tactile sensing on soft robot grippers, propose a novel design evaluation metric for vision-based tactile sensing, and a two-stage optimization process.

4.1 Design Space Description

We introduce the physical design of the robot in Chapter 2 and discuss optimization to improve the tactile sensing performance of our robot gripper. Due to specific design requirements and constraints, we keep the shape and material of each component the same. The sensing surface's range is determined by the field of view of the wide-angle camera we used. Because of the limitation of the field of view, we could not cover the entire silicone surface, we place the camera closer to the tip of our robot gripper because contacts are more likely to happen at the tip of the robot gripper than at the base. During the optimization, we will fix the camera's pose since it's already optimal. Now, we have a rectangle region for tactile sensing with three sides suitable for connecting optical fiber (Optical fibers are not able to be placed inside the silicone gel layer but could be attached through the 3D printed structure as shown in the figure 4.1 (a)). We evenly distribute optical fibers along each side and tilt each optical fiber a slight angle θ towards the sensing surface to provide as uniform illumination

as possible, shown in figure 4.1 (b). We will fix the number of optical fibers and their positions and tilting angle θ to reduce the complexity of optimization. Still, we could choose which of the three colored light sources (Green LEDs, Blue LEDs, Red LEDs) every optical fiber connects to and light direction ϕ.



(a) (b)

Figure 4.1: Schematic drawing of Lighting configuration. (a) is the actual robot optical fiber connections and we visualize optical fibers directions in three views on (b)

4.2 Design Evaluation

In Chapter 2, we review the theory of vision-based tactile sensing and its application in our context. In short, vision-based tactile sensing could be viewed as a function f from RGB color space to a normal vector, and we use various techniques, including lookup tables and neural network models to closely approximate this function in practice. Here, we define the evaluation metric function $\mathcal{M} : f \to R$ to measure the sensing performance. To formulate metric function $\mathcal{M}$, we need to look back to the design space and theory of light transportation. Given that our robot is in a steady state with a constant shape and assuming that the material of the sensing surface is Lambertian with albedo σ, for some position x on the sensing surface with a normal vector $n(\theta_n, \phi_n)$, we have the following radiance contribution from every light source

L_i with incident direction $\omega_i(\theta_i, \phi_i)$ in the same spherical coordinate:

$$L = \sigma \sum_i L_i \omega_i \cdot n \tag{4.1}$$

Here, we only consider situation where $\omega_i \cdot n > 0$.Converting ω_i and n back to Cartesian coordinate, We can calculate $\omega_i \cdot n$ by

$$\omega_i \cdot n = \begin{bmatrix} \sin(\theta_i)\sin(\phi_i) \\ \sin(\theta_i)\cos(\phi_i) \\ \cos(\theta_i) \end{bmatrix} \cdot \begin{bmatrix} \sin(\theta_n)\sin(\phi_n) \\ \sin(\theta_n)\cos(\phi_n) \\ \cos(\theta_n) \end{bmatrix} = \sin(\theta_i)\sin(\theta_n)\cos(\phi_i - \phi_n) + \cos(\theta_i)\cos(\theta_n) \tag{4.2}$$

Substitute 4.2 into 4.1, we have

$$\begin{aligned} L &= \sigma \sum_i L_i (\sin(\theta_i)\sin(\theta_n)\cos(\phi_i - \phi_n) + \cos(\theta_i)\cos(\theta_n)) \\ &= \sigma \sin(\theta_n) \sum_i L_i \sin(\theta_i)\cos(\phi_i - \phi_n) + \sigma \cos(\theta_n) \sum_i L_i \cos(\theta_i) \end{aligned} \tag{4.3}$$

From our physical design, $\cos\theta_i$ for each of the light sources are kept small because we want the lighting direction to be as parallel to the sensing surface as possible but also allow enough illumination to reach the sensing surface, so we could further approximate the above equation by neglecting the last term.

$$L = \sigma \sin(\theta_n) \sum_i L_i \sin(\theta_i)\cos(\phi_i - \phi_n) \tag{4.4}$$

Equation 4.4 essentially tells us how the color forms from a surface normal vector given the lighting condition and the normal evaluation function f is exactly the inverse function of this that it predicts a normal vector from color given current lighting condition. Therefore, a good function f is required to be one-to-one mapping so that each color could be correctly mapped to exactly one normal vector and maximize $f(\Delta n) \forall \Delta n$ to enhance robustness against noise. However, The nonlinearity of the above function and parameter changes due to the robot's bending state makes it hard to find an analytical optimization of f based on equation 4.4 under all robot conditions. Therefore, we use numerical sampling methods to estimate if function f

likely is a one-to-one mapping.

A naive solution will be to sample enough normal vectors $n_j(\theta, \phi)$ and check if the colors corresponding to the sampled normal vectors are different from each other. We could treat color as an integer space $[0, 255]^3$ and use Euclidean distance to describe how different colors are. Also, we need to penalize large variances in color differences because we want large color differences for all pairs of normal vectors. Thus, we arrive at our initial metric function

$$\mathcal{M} = \overline{d(c_i, c_j)} - \sigma(d(c_i, c_j)), \forall i, j \in \{0, \cdots, k\}, i \neq j \tag{4.5}$$

, where c_i, c_j are the RGB color corresponding to sampled normal vector n_i, n_j, and we uniformly sampled k normal vectors.

However, the above sampling method requires a lot of samples and computation per test, which is not desired for a quick evaluation metric considering our design space. To simplify the metric, we want to understand the effect of ϕ_n and θ_n on the normal estimation function f. From equation 4.4, the radiance response can be rewritten with $L = af(\theta_n)f(\phi_n)$, $f(\theta_n) = \sin(\theta_n)$. Because normal vectors always point from the sensing surface towards the camera, $\theta_n \in [0, \pi/2)$, making $f(\theta_n)$ a monotonous function that only acts on the magnitude of L. On the other hand, $f(\phi_n)$ contains all color phase transforms and nonlinearity of L. Here, we approximate the intensity of color with its norm in the integer space $[0, 255]^3$ and sample k normal vectors with equally spaced $\phi_i = \frac{i}{k}2\pi, i = 0, \cdots, k-1$ and a θ_i. We separate the metric function into two parts that measure changes in radiance response by θ_n and ϕ_n respectively. The metric function on θ_n evaluates the color intensity of each normal vector, and the metric function on ϕ_n evaluates the normalized color changes between nearby normal vectors. For both cases, metric values should also be evenly across different normal vectors so we penalize the variance of the metric values. In summary, we define our metric function as follows:

$$\begin{aligned} \mathcal{M} &= \mathcal{M}_{\theta_n} + \lambda \mathcal{M}_{\phi_n} \\ &= \overline{||c_{n_i}||} - \sigma(||c_{n_i}||) + \lambda(\overline{d(\hat{c}_{n_i}, \hat{c}_{n_{i+1}})} - \sigma(d(\hat{c}_{n_i}, \hat{c}_{n_{i+1}}))), \\ &\forall i \in \{0, \cdots, k-1\} \end{aligned} \tag{4.6}$$

where $\hat{c} = \frac{c}{||c||}$, $n_i = (\theta, \phi_i)$, $\phi_i = \frac{i}{k}2\pi, i = 0, \cdots, k-1$, λ is a scale factor to weight two metric functions equally.

In the simulation test, we extract color and normal vector pair by simulating robot contacting with a 10-faced pyramid-shaped indentor. Each face of the pyramid-shaped indentor provides a few pixels of colors corresponding to the same normal vector so that we could average these pixels' color to reduce any noise or error introduced by the simulator. We assume that the size of the indentor is small enough that the slight positional shift of the normal vector on each face of the indentor could be neglected. We sample 15 locations uniformly across the sensing surface with 3mm apart and 3 robot bending angles: $0, 45°, 90°$. Similar to how we define the metric function for sensing performance at some location, we define the score function for the entire lighting design as

$$\mathcal{S} = \overline{\mathcal{M}(p, b)} - (\sigma(\mathcal{M}(p) + \sigma(\mathcal{M}(b)), \forall p, b \tag{4.7}$$

, where p represents the position and b represents the bending angle.

4.3 Design Space Exploration

Now that we have our soft robot gripper modeled in the simulation and a metric that describes the sensing performance of our illumination design, we could finally proceed to the optimization stage. As stated in the section 4.1, we have two classes of parameters to optimize: color choice of the light source and direction of the light source. In the experiment, we observe that changes in the light source direction do not have as much effect on sensing performance as color choice because of our strong diffusion layer. Also, discrete parameter optimization is hard to treat with the same optimization technique as continuous parameters. Therefore, we split the optimization stage into two sub-stages. We first select the colored LED light source for each optical fiber and then fine-tune the optical fiber's direction.

4.3.1 Grid Search on Color Choices

In grid search, we iterate all possible combinations of color choices to find out the best color choice. However, it means that we will generate simulation results for 3^{20} combinations and evaluate every single design, which is not feasible given our computational resources and time. Here, we use two tricks to reduce the work. Leveraging the superposition of light, we render each simulation with only one LED on. While testing on a specific color choice, we superpose the rendering result by adding 20 images with the correct color layout. By doing so, we reduce the number of rendering from 3^{20} down to 3×20. Also, we notice that groups of neighboring several optical fibers with the same colored light source tend to have a high metric value than all neighboring optical fibers that have different colored light sources. It is largely because different colored light from the same direction creates ambiguity and violates the principle of photometric stereo. Therefore, we propose to group the two nearby optical fibers with the same color light source as shown in fig 4.2.

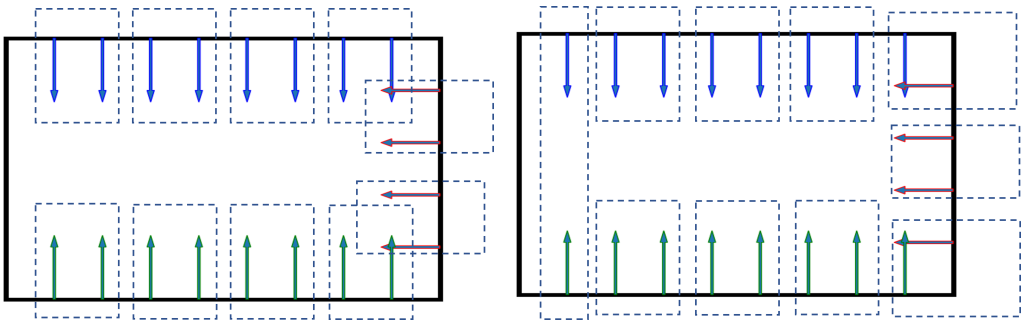


Figure 4.2: The two possible groupings for the grid search. To group two neighboring optical fibers together, there are two possible groupings layout as shown in (a) and (b) respectively

We could clearly find that optical fibers with the same colored light source tend to form large groups (> 2). Although we have obtained the best configuration under all groups of 2 cases (which automatically contains all even number grouping cases) and we show that large grouping has a positive impact on the sensing performance, we failed to prove that our illumination configuration success in cases contains any large odd number (≥ 3) groupings. Therefore, we change the color of the two optical

fibers' light at the boundary of two neighboring groups to ensure we didn't miss any odd number grouping.

4.3.2 Continuous Optimization of Light Direction

For continuous optimization of light directions, we jointly optimize 20 parameters $x \in [0, \pi]$. Because our simulation method is difficult to differentiate and non-convex, we use a stochastic numerical optimization technique called covariance matrix adaptation evolution strategy (CMA-ES) which is widely applied to solve this type of problem [22]. It iteratively updates the mean and covariance matrix of a multivariate normal distribution for searching the optimum by sampling random points in parameter space based on the distribution and estimating the mean and covariance matrix of the next-generation search distribution. For our continuous optimization, we run and evaluate the simulations with parameters sampled by CMA-ES for each generation until the changes in parameters are small enough. Although CMA-ES is particularly good at optimizing ill-conditioned functions like ours, it still takes a very long time to converge if we optimize all 20 optical fiber directions at the same time. Therefore, we group optical fibers with the same colored light source on the same side of the robot together and force them to share the same direction θ. The optimization is able to converge within 20 generations and the result is shown in Chapter 5.

Chapter 5

Results and Discussion

In this Chapter, we present our optimization result and provide intuitive insight into how our optimization improves the design.

5.1 Baseline Evaluation

We choose the lighting configuration in figure 5.1 as our baseline configuration. In this configuration, optical fibers with the same colored light source are placed on the same side of the robot and oriented normally to the surface, which is adopted in many gelsight-based tactile sensors [11]. This configuration exhibits a strong performance where all three colors of lights are presented from very distinct directions, shown on the right side of the heat maps in figure 5.1. For all the normal vectors towards the tip, it always receives at least two colored lights: either red and blue or red and green. All incidental light of the same color points in the same direction and the direction of lights of different colors are perpendicular to each other.

However, as the intensity of the tip light gradually diminishes towards the bottom of the robot, the sensing performance also drops down. Furthermore, we also find that when the robot bends at a larger angle, the tip light source diminishes more quickly towards the bottom of the robot. This configuration, although giving a high score at the tip location, exhibits significant variance in the spatial position and bending state of the robot. This pattern could be found in 5.1 and 5.2. Considering grasp contact could happen at any section of our robot gripper and usually happens

when the robot is in the actuated state (bending angle $> 0°$), the large variance in this design reduces the overall performance of the robot. Hence, we are interested in finding a lighting configuration that could balance the overall performance and variance in spatial distribution and bending states.

5.2 Color Choice Search Result

Compared to the baseline configuration, the asymmetric light source color choice provides both the tip and base section of the sensing surface with lights of three different colors. For positions near the tip, the sensing surface receives blue light from the tip and right side, red light from the left side, and green light from the right side near the base; For positions near the base, the sensing surface receives blue light from the left side, red light from the left side near the tip, and green light from the right side. One of the main differences in lighting conditions close to the base is that the third color light source (which is red in this case) comes from optical fibers at the side of the robot, which is closer than the optical fibers from the tip in the baseline condition. It overcomes the two shorting-comings of the baseline configuration. Because the third light source is closer, the sensing surface close to the base has a much stronger presence of the third colored light than the baseline configuration. Additionally, the sensing performance is much more robust to the changes in robot bending angles because the relative displacement of optical fibers to the sensing location is also related to the distance between them. Comparing heat maps in figure 5.1, the variance in both spatial and the robot bending state reduces notably.

Although the lighting configuration from grid search provides the presence of all three colored lights throughout the sensing surface, it fails to angle lights of different colors in distinct directions and balance the distribution of the three lights for all normal vectors. For instance, as shown in figure 5.2, positions close to the right side and the base have blue and red-colored light from similar directions; positions at the center of the sensing surface have blue lights from both the left side close to the base and the right side close to the tip; and positions at the right side near the tip receive a dominated blue light from both right side and the tip with little green light. All these examples violate the design principles summarized in Chapter 2, thus scoring

lower than other positions. Because most of the problem involves lighting directions, we expect that changing the light direction could help address the problem.

5.3 Light Direction Optimization Result

After optimization, metric values of almost all testing positions under all robot bending states increase marginally in figure 5.1. The main reason that changing lighting directions does not have a large impact on the metric values is that the silicone diffusive layer flattens out the radiance distribution of light from optical fibers. Furthermore, our metric for measuring sensitivity to θ_n in equation 4.6 penalizes any decrease in the light intensity received at the sensing surface while pointing optical fibers at the sensing surface gives the highest light intensity. Thus, tilting optical fibers too much away from the lighting configuration in section 5.2 decreases the overall performance.

Although large angle changes are not in favor, tilting optical fibers with certain small angles does improve the design as shown in the heat maps in figure 5.1. As stated in the previous section, one of the problems in the previous design is that incident light directions of different colors are very close to each other. Thus, the optimization result increases the angle between the optical fibers of different colored light sources at the same edge of the robot, for example, the blue and red light directions at the left side and the green and blue light directions at the right side. Another problem with the previous design is the dominant blue light over most normal vectors with less coverage of red and green light in the corner of the right side close to the tip. Now, the green light direction is pointed towards the tip so that more green light is able to reach the corner.

5.4 Indentation Comparison

For this experiment, the robot contacts a bolt under the baseline lighting design and optimized lighting design to demonstrate the improvement by our optimization methods. The results are shown in figure 5.3. Because the largest improvement of our optimized design to the baseline design is at the sensing area close to the base

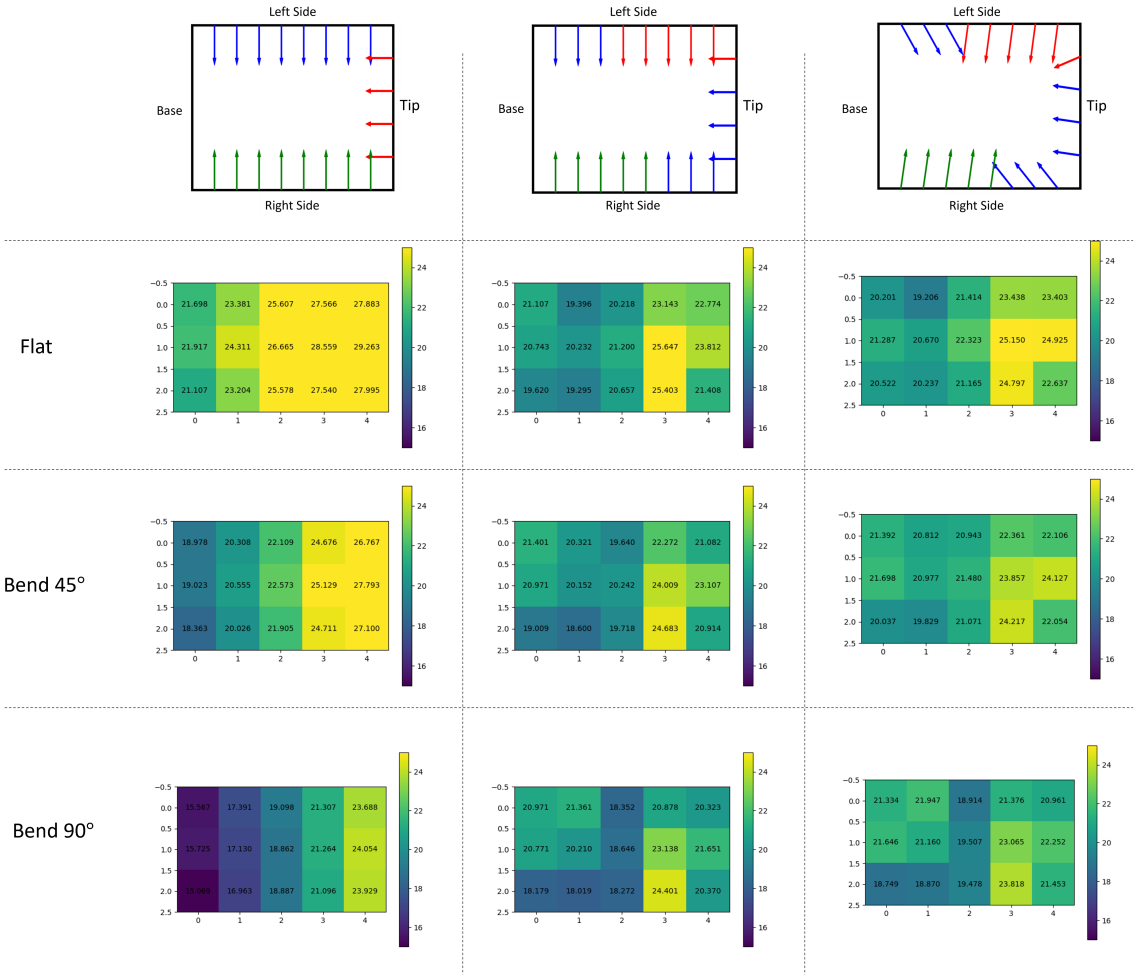


Figure 5.1: The heatmap of metric values at 15 sampling locations comparisons for the baseline configurations and configurations after each stage of the optimization. The first row is the scheme of the three lighting configurations: the baseline configuration, the configuration after the color choice search, and the configuration after light direction optimization. Each row shows the metric values at flat, 45°, 90° robot bending angles and each column corresponds to each light configuration.

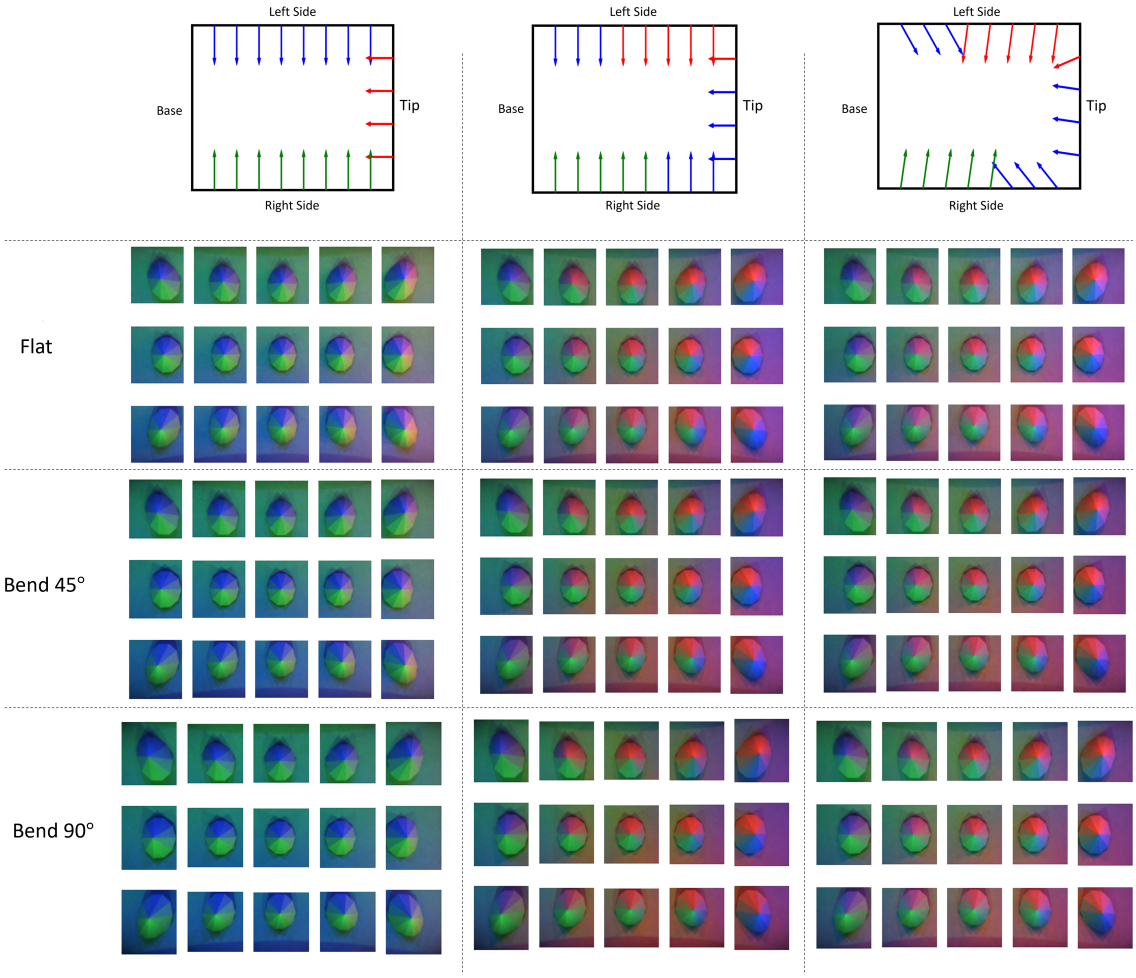
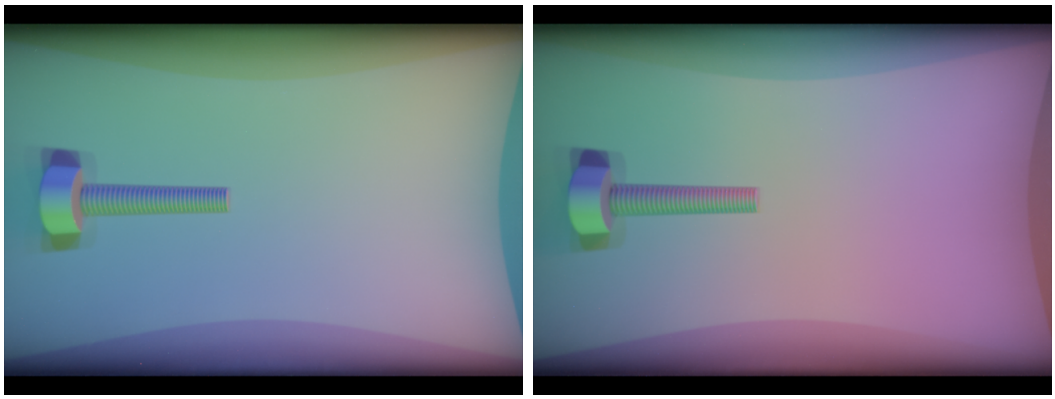
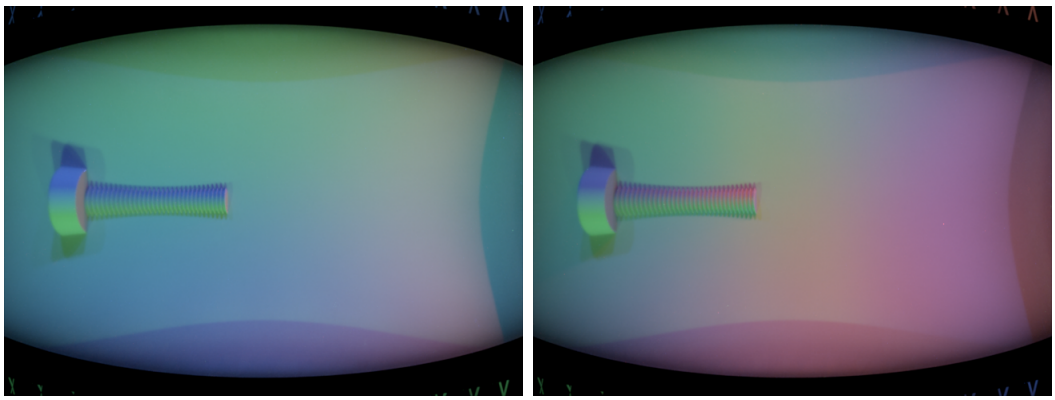


Figure 5.2: The image patches of indentation at 15 sampling locations comparisons for the baseline configurations and configurations after each stage of the optimization. The first row is the scheme of the three lighting configurations: the baseline configuration, the configuration after the color choice search, and the configuration after light direction optimization. Each row shows the image patches at flat, 45°, 90° robot bending angles and each column corresponds to each light configuration.

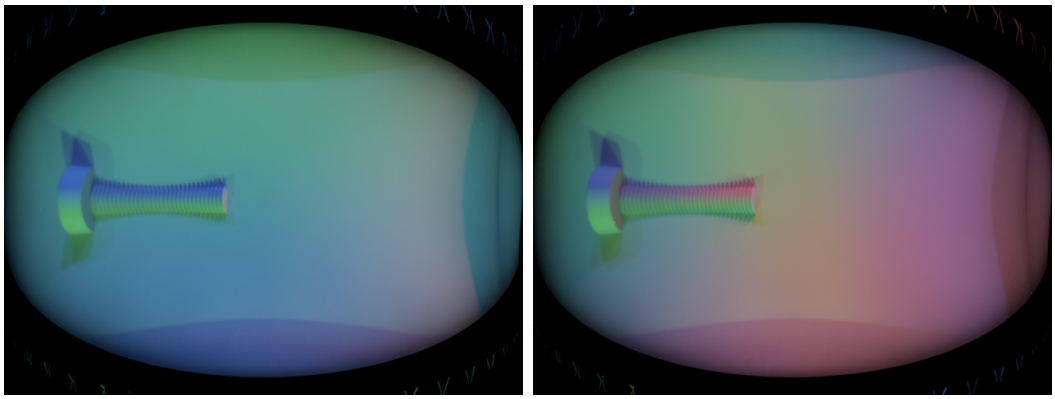
of the robot, we indent the bolt close to the base of the robot, which is shown on the right half of the images. In general, good lighting design could visualize sharp changes in the color at the sharp edges, thus we could use that to qualitatively judge the performance of the lighting system. For our baseline configuration in the first column, the edges closer to the tip are more visible than the edges closer to the base, and the edges are more visible when contacted by the robot in the flat bending state edges than when contacted by the robot in the bending state. However, the difference is much less noticeable in the optimized design in the second column. This is aligned with our analysis of the metric value distribution in the previous section that the baseline configuration has more variance of the performance in locations and robot bending states. Comparing designs under the same robot bending state, edges are generally sharper with optimization lighting configuration than the baseline configuration.



(a) (b)



(c) (d)



(e) (f)

Figure 5.3: The simulation results of robot grasping a bolt with baseline lighting design and optimized lighting design under different robot bending states. The first column (a,c,e) lists the results from baseline lighting design and the second column (b,d,f) lists the results from optimized lighting design. Each row represents one bending state of the robot. From top row to bottom row are the results of bending angle $0°, 45°, 90°$

Chapter 6

Conclusions and Future Directions

In this thesis, we present a physics-based optical simulation pipeline to guide the design of vision-based tactile sensing on a soft robot finger. Our simulation pipeline enables efficient iteration cycles and automatic optimization of design parameters compared to traditional trial-and-error methods, reducing the turnaround time by eliminating the manufacturing time. By leveraging physics-based rendering, the simulation tool provides highly accurate optical modeling of the robot finger and is shown the performance improvements in tactile sensing in the simulation are transferrable to the real-world robot finger. We also introduce a numerical metric to test tactile sensing performance at different robot actuation statuses and contact locations to evaluate designs. During the optimization, we perform a grid search on possible combinations of light color choices for optical fibers and apply the covariance matrix adaptation evolution strategy (CMA-ES) as a numerical optimization method to iteratively fine-tune the directions of the optical fibers. We compare the tactile sensing performance of the optimized design with the baseline design and demonstrate an improvement.

6.1 Exploration of Other Parameters

In this work, we specifically make some assumptions and fix a few design parameters for either simplification or physical design constraints. However, some design parameters are not trivial and have interesting effects on the sensing performance. One assumption

we made is that the indentation depth is small enough to neglect its impact on extracting the normal vector. However, the robot often grasps objects with sharp structures that could indent deep into the sensing surface. While our lighting system is designed to emit light almost parallel to the sensing surface, the large indentation depth could cast a shadow around the indentation. For example, the large bolt head shown in figure 5.3 casts shadows in both the left and right sensing surface. These shadows could affect the normal estimation process and create undesired artifacts around the indentor. This effect is not currently captured in our metric function but it could affect the quality of the final 3D reconstruction result.

6.2 Robustness Evaluation of Vision-based Tactile Design

In Chapter 2, we discussed how to improve tactile sensing robustness against noise and error from image formation. However, we discover that there are numerous additional errors and noise that arise during the manufacturing process, and these could be significant. For instance, when connecting optical fibers to colored light sources, which are colored LED diodes in our case, the I-V curves of these sources can vary slightly, leading to differences in intensity. Also, the LED diodes do not exactly follow the intensity profile described in the datasheet, leading to intensity differences. In Chapter 3, we assume the end of the optical fibers are perfect smooth surfaces that are perpendicular to the optical fiber's direction so that the light radiance distribution follows the normal distribution. However, achieving such a clean-cut surface is challenging, and as a result, the outgoing radiance distribution of the optical fiber may not be a normal distribution. We also found that it is extremely hard to manufacture a uniform scattering material for the diffusive layer. Any unevenness in the density will cause the diffusive layer to show different optical properties. All these errors happened on the manufacturing side impact the final sensing performance of vision-based tactile sensors.

One possible solution to address the impact of random manufacturing noises is to incorporate them into the design evaluation process. This can be achieved by introducing random offsets to the parameters in the optical simulation, such

as light distribution, scattering parameters, and surface smoothness, during the optimization process. By doing so, the final design will be more robust to errors from these parameters. However, this approach is only suitable for iterative optimization methods and may not be suitable for deterministic methods like grid search.

Alternatively, we can quantify the impact of small offsets on the overall sensing performance. Similar to taking the partial derivative, we can identify the most sensitive parameter to the sensing performance and mitigate its impact. By focusing on the most sensitive parameter, we can potentially reduce the impact of manufacturing noises on the overall sensing performance.

Bibliography

[1] J. Walker, T. Zidek, C. Harbel, S. Yoon, F. S. Strickland, S. Kumar, and M. Shin, "Soft robotics: A review of recent developments of pneumatic soft actuators," in *Actuators*, vol. 9, p. 3, MDPI, 2020. 1

[2] J. Guo, Y. Sun, X. Liang, J.-H. Low, Y.-R. Wong, V. S.-C. Tay, and C.-H. Yeow, "Design and fabrication of a pneumatic soft robotic gripper for delicate surgical manipulation," in *2017 IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 1069–1074, 2017. 1

[3] E. Navas, R. Fernández, D. Sepúlveda, M. Armada, and P. Gonzalez-de Santos, "Soft grippers for automatic crop harvesting: A review," *Sensors*, vol. 21, no. 8, 2021. 1

[4] D. Kim and Y.-L. Park, "Contact localization and force estimation of soft tactile sensors using artificial intelligence," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7480–7485, 2018. 1

[5] L. Massari, E. Schena, C. Massaroni, P. Saccomandi, A. Menciassi, E. Sinibaldi, and C. M. Oddo, "A machine-learning-based approach to solve both contact location and force in soft material tactile sensors," *Soft Robotics*, vol. 7, no. 4, pp. 409–420, 2020. PMID: 31880499. 1

[6] W. Yuan, R. Li, M. A. Srinivasan, and E. H. Adelson, "Measurement of shear and slip with a gelsight tactile sensor," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 304–311, 2015. 1

[7] S. Suresh, Z. Si, J. G. Mangelson, W. Yuan, and M. Kaess, "Shapemap 3-d: Efficient shape mapping through dense touch and vision," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 7073–7080, 2022. 1

[8] R. L. Truby, R. K. Katzschmann, J. A. Lewis, and D. Rus, "Soft robotic fingers with embedded ionogel sensors and discrete actuation modes for somatosensitive manipulation," in *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*, pp. 322–329, 2019. 1.1

[9] T. J. Pannen, S. Puhlmann, and O. Brock, "A low-cost, easy-to-manufacture, flexible, multi-taxel tactile sensor and its application to in-hand object recogni-

tion," in *2022 International Conference on Robotics and Automation (ICRA)*, p. 10939–10944, IEEE Press, 2022. 1.1

[10] Y. She, S. Q. Liu, P. Yu, and E. Adelson, "Exoskeleton-covered soft finger with vision-based proprioception and tactile sensing," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10075–10081, 2020. 1.1

[11] S. Q. Liu and E. H. Adelson, "Gelsight fin ray: Incorporating tactile sensing into a soft compliant robotic gripper," in *2022 IEEE 5th International Conference on Soft Robotics (RoboSoft)*, pp. 925–931, 2022. 1.1, 5.1

[12] R. J. Woodham, "Photometric method for determining surface orientation from multiple images," *Optical engineering*, vol. 19, no. 1, pp. 139–144, 1980. 2.1

[13] C. Kelemen, L. Szirmay-Kalos, G. Antal, and F. Csonka, "A simple and robust mutation strategy for the metropolis light transport algorithm," *Comput. Graph. Forum*, vol. 21, 09 2002. 3.2

[14] W. Jakob, "Mitsuba renderer," 2010. http://www.mitsuba-renderer.org. 3.2

[15] T. Tamir, G. Griffel, and H. L. Bertoni, *Guided-wave optoelectronics: device characterization, analysis, and design*. Springer Science & Business Media, 2013. 3.3.1

[16] A. W. Snyder and W. R. Young, "Modes of optical waveguides," *JOSA*, vol. 68, no. 3, pp. 297–309, 1978. 3.3.1

[17] I. Cree, "Data sheet for clm2d-gpc/bpc: Plcc4 green & blue smd leds." https://www.mouser.com/datasheet/2/723/1410_CLM2D_GPC_BPC-2257056.pdf. 3.3.1

[18] I. Gkioulekas, S. Zhao, K. Bala, T. Zickler, and A. Levin, "Inverse volume rendering with material dictionaries," *ACM Trans. Graph.*, vol. 32, nov 2013. 3.3.1

[19] J. Gu, S. Nayar, E. Grinspun, P. Belhumeur, and R. Ramamoorthi, "Compressive structured light for recovering inhomogeneous participating media," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, 05 2012. 3.3.1

[20] S. G. Narasimhan, M. Gupta, C. Donner, R. Ramamoorthi, S. K. Nayar, and H. W. Jensen, "Acquiring scattering properties of participating media by dilution," *ACM SIGGRAPH 2006 Papers*, pp. 1003–1012, 2006. 3.3.1

[21] A. Agarwal, T. Man, and W. Yuan, "Simulation of vision-based tactile sensors using physics based rendering," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–7, 2021. 3.3.1

[22] N. Hansen, S. D. Müller, and P. Koumoutsakos, "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es)," *Evolutionary Computation*, vol. 11, no. 1, pp. 1–18, 2003. 4.3.2