Perception amidst interaction: spatial AI with vision and touch for robot manipulation

Sudharshan Suresh

CMU-RI-TR-24-06

February 2024

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

Thesis Committee

Michael Kaess	Carnegie Mellon University (**chair**)
Nancy Pollard	Carnegie Mellon University
Shubham Tulsiani	Carnegie Mellon University
Mustafa Mukadam	FAIR, Meta
Alberto Rodriguez	Boston Dynamics / MIT

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.

Keywords: robot manipulation; tactile sensing; SLAM; neural fields

Abstract

Robots currently lack the cognition to replicate even a fraction of the tasks humans do, a trend summarized by Moravec's Paradox. Humans effortlessly combine their senses for everyday interactions—we can rummage through our pockets in search of our keys, and deftly insert them to unlock our front door. Before robots can demonstrate such dexterity, they must first exhibit spatial awareness of the objects they manipulate. Specifically, object pose and shape are important quantities for downstream planning and control. The status quo for in-hand perception is restricted to the narrow scope of tracking known objects with vision as the dominant modality. As robots move out of instrumented labs and factories to cohabit our spaces, it is clear that a missing piece is generalizable *spatial AI*.

Often overlooked is tactile sensing, which provides a direct window into robot-object interaction, free from occlusion and aliasing. With hardware advances like vision-based touch, we now have situated yet detailed information to complement cameras. However, interactive perception is intrusive—the act of sensing itself perturbs the object. *Can we robustly estimate object shape and pose online from a stream of multimodal robot manipulation data?*

In this thesis, I study the intersection of simultaneous localization and mapping (SLAM) and robot manipulation. More specifically, I look at: (1) spatial representations for object-centric SLAM, (2) tactile perception and simulation, and (3) combining learned models with online optimization. First, I show how factor graphs fuse touch with physics-based constraints for SLAM in planar manipulation (Chapter 2). Next, I present a schema for online shape learning from visuo-tactile sensing (Chapter 3). I then demonstrate a learned tactile representation for global localization via touch (Chapter 4). Drawing upon the above efforts, I culminate with unifying vision, touch and proprioception into a neural representation for SLAM during in-hand manipulation (Chapter 5).

Acknowledgments

I first thank my advisor, Michael Kaess, for his thoughtful guidance, steadfast support, and utmost patience through the years. It's been a pleasure working with him ever since I was a Masters student at CMU. Through his guidance, I learned how to tackle a research statement from inception to dissemination. He has a deep knowledge of perception and optimization literature, and is quick to draw parallels between seemingly disparate fields. He's given me the freedom to explore my interests, shape my research journey, and forge lasting collaborations. He's always been supportive of my goals, while shielding me from anything that isn't research.

I thank my thesis committee for opening doors, asking the right questions, and guiding my vision. Mustafa Mukadam has been a second advisor to me through the latter half of my PhD. I'll look back fondly at our weekly meetings, where we would fill up the whiteboard at the FAIR Pittsburgh office. It's not just our theoretical discussions; he taught me how to spotlight my work and make it accessible for larger research community. I benefited immensely from his knack of bringing together collaborators and tapping into resources across industry and academia.

Alberto Rodriguez has been a kind, thoughtful voice with utmost expertise in the problems I want to solve. During my first year, in the midst of the pandemic, I cold emailed him about my desire to further his lab's work. This evolved into a fruitful collaboration, and I stood to gain from the cross-pollination of ideas between our labs ever since. Nancy Pollard was crucial in shaping my research ever since I started my PhD. My discussions with her gave me confidence in our research direction, while keeping me grounded on the challenges of real-world manipulation. Shubham Tulsiani has always asked the pertinent questions, helped me zoom out on the larger picture, and has been a valuable resource for 3D perception.

It takes a village to write a dissertation—I start by thanking the members of the robot perception lab: Allie, Akash, Akshay, Andrew, Dan, Dominic, Eric Dexheimer, Eric Westman, Joe, Josh, Ming, Mohamad, Monty, Nora, Paloma, Wei, and many others. The lab is forged in the spirit that no matter what robot we work with, the underlying optimization problems are similar. I'll look back on the summer reading groups, weekly presentations, conference pushes, and erstwhile watercooler sessions. Wenzhen Yuan and Zilin Si, for introducing me to the world of vision-based touch and being such gracious collaborators. Maria Bauza, for her advise through the years on research and career, and Peter Yu, who set in motion my first PhD research problem. I am indebted to Red Whittaker, Rachel Burcin, and John Dolan—without the RISS program I wouldn't be here in the first place.

This thesis would not be possible without the folks at FAIR: Roberto Calandra, for his acute understanding of touch, manipulation, and scientific writing—I'm

grateful to stand on the shoulders of the work he has pioneered. Joe Ortiz and Haozhi Qi, for being key collaborators in my research—I've learned so much about perception and reinforcement learning from them. Stuart Anderson, for teaching me how to dissect, visualize, and proof-of-concept a seemingly hard research problem. Tingfan Wu and Mike Lambeta, for their steadfast support on hardware and tactile sensing during my summer internship. Luis Pineda, for guiding me through the nuts and bolts of environment management, version control, and remote deployment. Taosha Fan, for his knowledge of backend optimization, along with the rest of the Theseus team. Mrinal Kalakrishnan, Tess Hellebrekers, Dhruv Batra, Jitendra Malik, and Franziska Meier, for being champions of my research.

I am incredibly thankful for all my friends across time zones—in Pittsburgh, the USA, and my wonderful hometown of Chennai, India. To Abhijat for being my housemate for as long as I've been in Pittsburgh; Tanmay and Raunaq for wonderful chai and conversations; Monty for putting me on literal thin ice during our annual Panther Hollow Lake ice skate; Pragna for bringing the social fabric together by organizing every gathering I show up to; Michelle, Ben, and Nishchay for coming outside with me in skis and/or running shoes; Cherie and Thomas for long conversations over tea that put us way off schedule; Vishwesh, Athindra, and Medhini for being the strongest of supports from long-distance and keeping me happy through phone calls and trips. The Indian graduate student band at CMU was a great group to unwind with in the semester, and we remain a tightly-knit to date. Pittsburgh holds a special place in my heart—I'll miss my runs in Frick and Schenley, pastries at Madeleine Bakery, and Lattes at Commonplace Coffee.

I'm filled with gratitude towards my fiancé, Varsha, for her undying support through the years. Research has its ups and downs and she's been with me through and through. I can't wait to see what the future holds for us. Finally and most importantly, I thank my family—Amma, Appa, Karthik, and the extended circle—who put my needs before theirs so I could lead a privileged life. They've always been a phone call away and placed all their trust in my extended education. It is their love and support that empowers me to follow my dreams.

I acknowledge funding support from the National Science Foundation, Hima and Jive Fellowship, and FAIR, Meta.

Contents

List of Figures

Chapter 1

Introduction

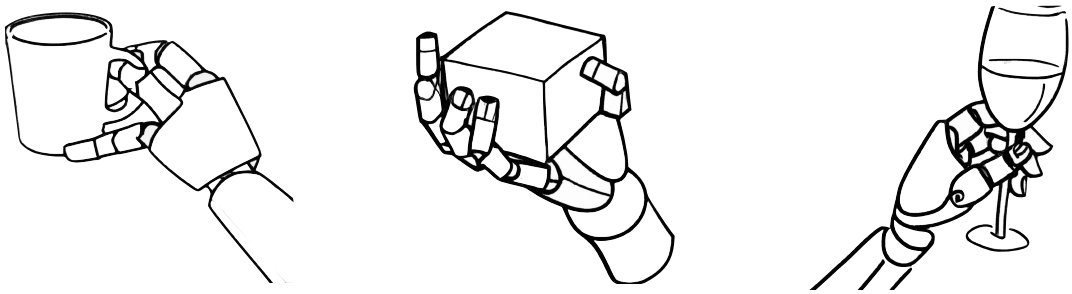


Figure 1.1: This thesis is towards developing the core competencies of embodied perception in robots. Interaction begets rich multimodal signals—can we leverage this data towards spatial understanding of objects?

Robots have dominated manufacturing supply chains ever since the Industrial revolution, and serviced the most inhospitable environments from Mars to Fukushima. Paradoxically, the largest leap we are yet to make is for robots to operate in own homes. Currently, robots fail to replicate even a fraction of the mundane tasks we perform, a trend summarized by Moravec's Paradox [Mor88]. Unstructured environments with apriori unknown objects present a diverse set of challenges across perception, planning, and control. In this thesis, we focus on the specific task of *geometric perception*—for embodied intelligence a robot must first understand its spatial relationship with respect to the manipuland.

Humans, however, achieve a broad range of everyday tasks through a combination of vision, touch, and proprioception. Even when blindfolded, we can locate, interact with, and perceive object geometries without global cues [KLM85; LL86; LK87]. Studies show we optimally fuse multimodal signals [HE07]—vision gives coarse global context, while touch gives precise local information. In practice, vision can locate a mug on the cluttered counter-top while touch can singulate the contours of the handle for a firm grasp. Despite these findings, the dominant percept in robot manipulation is still vision.

In computer vision and cognition, researchers tend to tolerate interaction rather than embrace it. This is at odds with researchers in manipulation, who regularly operate in regimes where self-occlusion is imminent—like rotating [Qi+22], re-orienting [Ope+18; Han+22; Che+23a], finger-gaiting [SH18], multi-finger enclosure [Bra+19], and sliding [Shi+17; CHR18; She+21]. Additionally, vision often fails in the real-world due to poor illumination, limited range, transparency, and specularity. Touch provides a direct window into these dynamic interactions, but a general technique for the underlying inference still remains an open question [Luo+17].

For robot manipulation knowledge of *object pose and shape* has been shown to be crucial to policy generalization [Ope+18; Ope+19; Han+22; Qi+23]. As opposed to end-to-end supervision [Yin+23; Guz+23b; Che+23a; Guz+23a], these methods require a persistent 3D representation of the object. However, the status quo for in-hand perception is currently restricted to the narrow scope of tracking known objects with vision as the dominant modality [Han+22]. Further, it is common for practitioners to sidestep the perception problem entirely, retrofitting objects and environments with fiducials [Ope+18; Ope+19]. To further progress towards general dexterity, it is clear that one missing piece is general, robust perception.

Alongside the challenges, there are contemporaneous advances in tactile sensing, rendering, and computer vision that make this an opportune time to pursue this direction. First, development of vision-based touch sensors [YDA17; Don+18; War+18; Als+19; Lam+20; Pad+20; Wan+21]—like the GelSight and DIGIT—provide high spatial acuity at an affordable price point. With a fingertip form-factor, their illuminated gel deforms on contact and the physical interaction is captured by an internal camera. When chained with robot kinematics, we obtain dense, situated contact that can be processed similar to natural camera images. Second, tactile simulation [Wan+22; AMY21; SY22] with realistic rendering enables practitioners to learn tactile observation models that transfer to real-world interactions [Wan+21; Sod+22b]. Third, the proliferation of online learning in 3D perception [Ort+22; Suc+21; Zhu+22] sets us up to transfer these ideas towards SLAM for manipulation.

1.1 Problem statement

In this thesis, we focus online methods for object localization and shape reconstruction from robot manipulation data. We tackle the question: *can we estimate shape and pose online from multimodal robot manipulation data?* We operate under the constraints of robots in the wild: **(i)** causal perception *i.e.*, no access to future information, **(ii)** lack of fiducials and sophisticated motion capture, **(iii)** noisy, occluded multimodal sensing *i.e.*, vision, touch, and proprioception, **(iv)** apriori unknown objects.

Such a system can serve as a perception module for a robot manipulation stack, directly feeding into downstream planning and control. This is especially relevant in unstructured settings, such as homes, where robots encounter novel objects everyday and must continually learn from in-

teraction. Additionally, our open-source evaluation datasets bring down the barrier of entry for visuo-tactile perception research, with some early adopters [HBM23]. While active perception and perception driven planning are not covered in this thesis, we highlight relevant contributions in future work (Section 5.9).

The domain of our work—an intersection of simultaneous localization and mapping (SLAM) and manipulation—has been studied for over two decades. A first exemplar is from Moll and Erdmann [ME04], who reconstruct the shape and motion of an object rolled between robot palms, later reproduced with specialized sensors [Str+14; Lep+23]. Tactile SLAM has been thoroughly investigated for planar pushing due to its well-understood mechanics [YLR15]. The combination of vision and touch has been explored for reconstructing fixed objects [Wan+18; Smi+20; Che+23b] and tracking objects of known shape [YR18; Lam+19; Sod+21].

1.2 Thesis contributions

We divide our contributions by chapter, each tackling a specific aspect of the object perception problem for manipulation. They are ordered in increasing complexity of interaction tasks—planar pushing, discrete 3D interaction, forceful sliding, and in-hand rotation.

- **Chapter 2**: We consider the problem of estimation for planar contact interactions from tactile sensing. With just force-torque sensing at the distal end, we estimate both object shape and pose for a pusher-slider system. Pure tactile inference is challenging because, unlike vision, touch cannot directly provide global estimates of object model or pose. Fusing touch with physics-based constraints, we develop a factor graph optimizer that works in tandem with a learned shape representation. This work forms the motivation for the alternating shape/pose optimization in Chapter 5.

> **[Sur+21]** *Tactile SLAM: real-time inference of shape and pose from planar pushing* by Sudharshan Suresh, Maria Bauza, Peter Yu, Joshua Mangelson, Alberto Rodriguez, and Michael Kaess. Proc. IEEE Intl. Conf. on Robotics and Automation, ICRA, (Xi'an, China), May 2021. https://www.cs.cmu.edu/~sudhars1/tactile-slam

- **Chapter 3**: Here, we present a method towards online shape learning with vision and touch sensing. Using vision-based touch and RGB-D sensing, we fuse the resulting 3D data into an efficient implicit shape representation. We train an image to depth network for the GelSight in simulation, so as to give situated 3D pointclouds similar to RGB-D. We then incorporate the 3D measurements into our spatial graph structure for online inference of the object's SDF.

[Sur+22b] *ShapeMap 3-D: Efficient shape mapping through dense touch and vision* by Sudharshan Suresh, Zilin Si, Joshua Mangelson, Wenzhen Yuan, and Michael Kaess. Proc. IEEE Intl. Conf. on Robotics and Automation, ICRA, (Philadelphia, USA), May 2022. https://www.cs.cmu.edu/~sudhars1/shape-map

- **Chapter 4**: MidasTouch is a nonparameteric filtering framework for global localization of a robot finger on known object models. We introduce a tactile place recognition module that compresses DIGIT tactile images to learned embeddings to inform a Monte-Carlo filter. This outputs a multimodal pose distribution of the robot finger, that evolves from sliding interactions with these objects. With this, we open-source *YCB-Slide*, the largest such tactile perception dataset with sensor-object pose annotation.

[Sur+22a] *MidasTouch: Monte-Carlo inference over distributions across sliding touch* by Sudharshan Suresh, Zilin Si, Stuart Anderson, Michael Kaess, and Mustafa Mukadam. Proc. Conf. on Robot Learning, CoRL, (Auckland, NZ), Dec. 2022. https://suddhu.github.io/midastouch-tactile

- **Chapter 5**: We combine vision and touch sensing on a multifingered hand to estimate an objects pose and shape during in-hand manipulation. Our method, NeuralFeels encodes object geometry by learning a neural field online and jointly tracks it by optimizing a pose graph problem. We unify vision, touch, and proprioception to demonstrate SLAM for apriori unknown objects, and robust tracking of known objects. In our experiments, we present our robot with a novel object, and it infers and tracks its geometry through just interaction. We use a dexterous hand sensorized with commercial vision-based touch sensors and a fixed RGB-D camera. We release our evaluation dataset of 70 experiments, *FeelSight*, as a step towards benchmarking in this domain. Our results demonstrate that touch, at the very least, refines and, at the very best, disambiguates visual estimates during in-hand manipulation.

[Sur+23] *Neural feels with neural fields: Visuo-tactile perception for in-hand manipulation* by Sudharshan Suresh, Haozhi Qi, Tingfan Wu, Taosha Fan, Luis Pineda, Mike Lambeta, Jitendra Malik, Mrinal Kalakrishnan, Roberto Calandra, Michael Kaess, Joe Ortiz, and Mustafa Mukadam. *Under review*, arXiv preprint arXiv:2312.1346, Dec. 2023. https://suddhu.github.io/neural-feels

Chapter 2

Shape and pose from planar pushing

Tactile SLAM

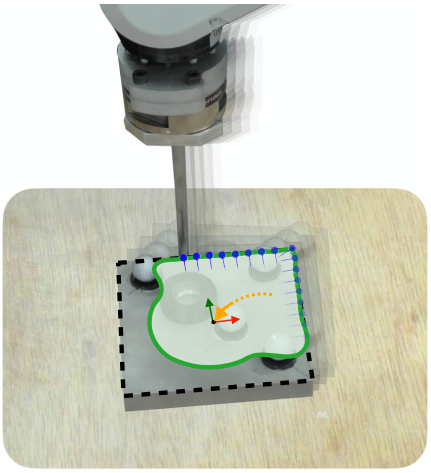


Figure 2.1: The shape and pose estimation problem in a pusher-slider system. A robot manipulator pushes along a planar object, while recording a stream of tactile measurements. Our method builds a shape contour in real-time as a Gaussian process implicit surface, and optimizes for pose via geometry and physics-based constraints. Figure shows tactile measurements (•), estimated motion (↓), estimated shape/pose (○), and ground-truth (- -).

For effective interaction, robot manipulators must build and refine their understanding of the world through sensing. This is especially relevant in unstructured settings, where robots have little to no knowledge of object properties, but can physically interact with their surroundings. Even when blindfolded, humans can locate and infer properties of unknown objects through touch [KLM85]. Replicating some fraction of these capabilities will enable contact-rich manipulation in environments such as homes and warehouses. In particular, knowledge of object shape and pose determines the success of generated grasps or nonprehensile actions.

While there have been significant advances in tactile sensing, from single-point sensors to high-resolution tactile arrays, a general technique for the underlying inference still remains an open

This chapter is adapted from the publication *Tactile SLAM: real-time inference of shape and pose from planar pushing* by Sudharshan Suresh, Maria Bauza, Peter Yu, Joshua Mangelson, Alberto Rodriguez, and Michael Kaess. In Proc. IEEE Intl. Conf. on Robotics and Automation, ICRA, (Xi'an, China), May 2021

question [Luo+17]. Visual and depth-based tracking have been widely studied [SNF15], but suffer from occlusion due to clutter or self-occlusions with the gripper or robot. We provide a general formulation of pure tactile inference, that could later accommodate additional sensing modalities.

Pure tactile inference is challenging because, unlike vision, touch cannot directly provide global estimates of object model or pose. Instead, it provides detailed, local information that must be fused into a global model. Moreover, touch is intrusive: the act of sensing itself constantly perturbs the object. We consider tactile inference as an analog of the well-studied simultaneous localization and mapping (SLAM) problem in mobile robotics [DB06]. Errors in object tracking accumulate to affect its predicted shape, and vice versa.

Central to this problem is choosing a shape representation that both faithfully approximates arbitrary geometries, and is amenable to probabilistic updates. This excludes most parametric models such as polygons/polyhedrons [YLR15], superquadrics [SB90], voxel maps [Dun+13], point-clouds [Mei+11], and standard meshes [Var+17]. Gaussian process implicit surfaces (GPIS) [WF07] are one such nonparametric shape representation that satisfies these requirements.

In this chapter, we demonstrate online shape and pose estimation for a planar pusher-slider system. We perform tactile exploration of the object via contour following, that generates a stream of contact and force measurements. Our novel schema combines efficient GPIS regression with factor graph optimization over geometric and physics-based constraints. The problem is depicted in Fig. 2.1: the pusher moves along the object while estimating its shape and pose.

We expand the scope of the batch-SLAM method by Yu et al. [YLR15] with a more meaningful shape representation, and real-time online inference. Our contributions are:

(1) Tactile SLAM that alternates between shape regression and incremental pose optimization,
(2) Implicit surface generation from touch using overlapping local Gaussian processes (GPs),
(3) Results from tactile exploration across planar object shapes in simulated and real settings.

2.1 Related work

SLAM and object manipulation

Our work is closely related to that of Yu et al. [YLR15], that recovers shape and pose from tactile exploration of a planar object. The approach uses contact measurements and the well-understood mechanics of planar pushing [LMT92; GRP91] as constraints in a batch optimization. Naturally, this is expensive and unsuitable for online tactile inference. Moreover, the object shape is represented as ordered control points, to form a piecewise-linear polygonal approximation. Such a representation poorly approximates arbitrary objects, and fails when data-association is incorrect. Moll and Erdmann [ME04] consider the illustrative case of reconstructing motion and

shape of smooth, convex objects between two planar palms. Strub et al. [Str+14] demonstrate the full SLAM problem with a dexterous hand equipped with tactile arrays.

Contemporaneous research considers one of two simplifying assumptions: modeling shape with fixed pose [Mei+11; DTG11; Mar+13; Yi+16; DHT19], or localizing with known shape [PK11; ZLT13; KPS15; BCR19; Sod+21; Sod+22a]. The extended Kalman filter (EKF) has been used in visuo-tactile methods [Heb+11; Iza+17], but is prone to linearization errors. At each timestep, it linearizes about a potentially incorrect current estimate, leading to inaccurate results. Smoothing methods [Kae+12] are more accurate as they preserve a temporal history of costs, and solve a nonlinear least-squares problem. These frameworks have been used to track known objects with vision and touch [YR18; Lam+19], and rich tactile sensors [Sod+21; Sod+22a].

Gaussian process implicit surfaces

A continuous model which can generalize without discretization errors is of interest to global shape perception. Implicit surfaces have long been used for their smoothness and ability to express arbitrary topologies [Bli82]. Using Gaussian processes [Ras03] as their surface potential enables probabilistic fusion of noisy measurements, and reasoning about shape uncertainty. GPIS were formalized by Williams and Fitzgibbon [WF07], and were later used by Dragiev et al. to learn shape from grasping [DTG11]. It represents objects as a signed distance field (SDF): the signed distance of spatial grid points to the nearest object surface. The SDF and surface uncertainty were subsequently used for active tactile exploration [DTG13; Li+16; Yi+16]. To our knowledge, no methods use GPIS alongside pose estimation for manipulation tasks.

Online GP regression scales poorly due to the growing cost of matrix inversion [Ras03]. Spatial mapping applications address this by either sparse approximations to the full GP [SG06], or training separate local GPs [KK13; SS20]. Lee et al. [Lee+19] propose efficient, incremental updates to the GPIS map through spatial partitioning.

2.2 Problem formulation

We consider a rigid planar object on a frictional surface, moved around by a pusher (see Fig. 2.1). The interaction is governed by simple contour following for tactile exploration. Given a stream of tactile measurements, we estimate the 2-D shape and object's planar pose in real-time.

Object pose: The object pose at the current timestep t is defined by position and orientation $\mathrm{x}_t = (x,\ y,\ \theta) \in SE\,(2)$.

Object shape: The estimated object shape is represented as an implicit surface $\mathcal{S} \in \mathbb{R}^2$ in the reference frame of x_t.

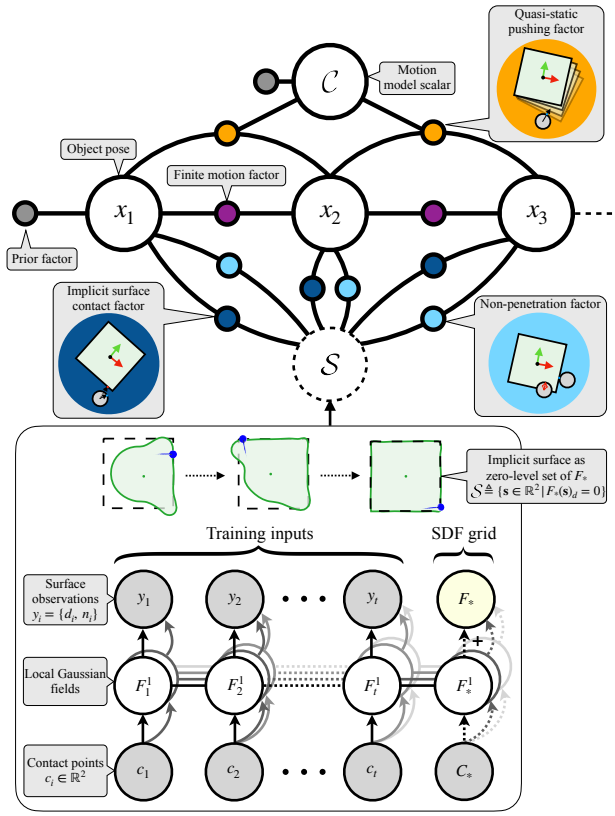


Figure 2.2: The combined formulation between the factor graph (Section 2.4) and GPIS (Section 2.3). **[top]** The graph shows the relationship between the variables to be optimized for (circles) and the factors that act as constraints (colored dots). **[bottom]** Our GPIS builds an implicit surface shape representation that is the zero level-set of GP potential function. Spatial partitioning with local GPs enables efficient regression.

Tactile measurements: At every timestep we observe:

$$z_t = \left\{ \underbrace{p_t \in \mathbb{R}^2}_{\text{pusher position}} , \underbrace{f_t \in \mathbb{R}^2}_{\text{force vector}}, \underbrace{\Theta_t \in \{0,1\}}_{\text{contact/no-contact}} \right\} \tag{2.1}$$

Pusher position is obtained from the robot's distal joint state, and force is sensed directly from a force/torque sensor. Contact is detected with a minimum force threshold, and the estimated contact point c_t is derived from knowledge of p_t, f_t and probe radius r_{probe}. This is consistent with the formulation in [YLR15], with the addition of direct force measurements f_t. For simplicity we consider a single pusher, but it can be expanded to multiple pushers or a tactile array.

Assumptions: We make a minimal set of assumptions, similar to prior work in planar pushing [YLR15; YR18; Lam+19].

- Quasi-static interactions and limit surface model [GRP91; LC91],
- Uniform friction and pressure distribution between bodies,
- Object's rough scale and initial pose, and
- No out-of-plane effects.

2.3 Shape estimation with implicit surfaces

2.3.1 Gaussian process implicit surface

A Gaussian process learns a continuous, nonlinear function from sparse, noisy datapoints [Ras03]. Surface measurements are in the form of contact points c_i and normals n_i, transformed to an object-centric reference frame. Given N datapoints, the GP learns a mapping $X \mapsto Y$:

$$F : \underbrace{\{c_{i_x},\ c_{i_y}\}^{i=1\cdots N}}_{X \in \mathbb{R}^2} \mapsto \underbrace{\{d_i, n_{i_x}, n_{i_y}\}^{i=1\cdots N}}_{Y \in \mathbb{R}^3} \tag{2.2}$$

$$\text{where } d \text{ represents signed-distance from object surface} \quad \begin{cases} d = 0, & \text{on surface} \\ d < 0, & \text{inside object} \\ d > 0, & \text{outside object} \end{cases} \tag{2.3}$$

The posterior distribution at a test point c_* is shown to be $F_* \sim \mathcal{GP}(\bar{F}_*, \sigma_*^2)$, with output mean and variance [Ras03]:

$$\begin{aligned} \bar{F}_* &= k_*^T \left(K + \sigma^2_{\text{noise}} I\right)^{-1} Y \\ \sigma_*^2 &= k_{**} - k_*^T \left(K + \sigma^2_{\text{noise}} I\right)^{-1} k_* \end{aligned} \tag{2.4}$$

where $K \in \mathbb{R}^{N \times N}$, $k_* \in \mathbb{R}^{N \times 1}$ and $k_{**} \in \mathbb{R}$ are the train-train, train-test, and test-test kernels respectively. We use a thin-plate kernel [WF07], with hyperparameter tuned for scene dimensions. The noise in output space is defined by a zero-mean Gaussian with variance σ^2_{noise}. While contact points condition the GP on zero SDF observations, contact normals provide function gradient observations [DTG11]. Thus, we can jointly model both SDF and surface direction for objects.

We sample the posterior over an M element spatial grid of test points C_*, to get SDF F_{*_d}. The estimated implicit surface $\mathcal{S}$ is then the zero-level set contour:

$$\mathcal{S} \triangleq \{\mathrm{s} \in \mathbb{R}^2 \mid F_*(\mathrm{s})_d = 0\} \tag{2.5}$$

The zero-level set $\mathcal{S}$ is obtained through a contouring subroutine on F_{*_d}. $\mathcal{S}$ is initialized with a circular prior and updated with every new measurement $\{\mathrm{c}_i, \mathrm{n}_i, d{=}0\}$. In Fig. 2.3 we reconstruct the `butter` shape [Yu+16] with a sequence of noisy contact measurements.

2.3.2 Efficient online GPIS

In a naïve GP implementation, the computational cost restricts online shape perception. Eq. (2.4) requires an $N \times N$ matrix inversion that is $O(N^3)$, and spatial grid testing that is $O(MN^2)$. We use local GPs and a subset of training data approximation for efficient online regression:

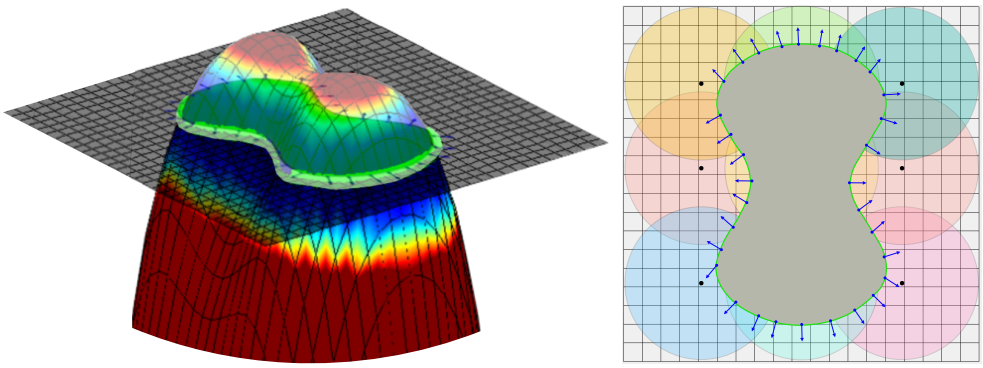


Figure 2.3: [left] Gaussian process potential function and its implicit surface (green) for noisy contact measurements on the `butter` shape [Yu+16]. The colormap shows spatial grid uncertainty. **[right]** The overlapping local GPs $F^1 \dots F^L$. Each GP is responsible for training and test points within its radius, and the overlapping regions ensure continuity in the shape contour.

Local GP regression: We adopt a spatial partitioning approach similar to [KK13; Lee+19; SS20]. The scene is divided into L independent local GPs $F^1 \dots F^L$, each with a radius r and origin (Fig. 2.3). Each F^i claims the training and test points that fall within r of its origin. The GPs effectively govern smaller domains ($N_{F^i} \ll N$ and $M_{F^i} \ll M$), and not the entirety of the spatial grid. At every timestep: (i) a subset of local GPs are updated, (ii) only the relevant test points are resampled. Kim et al. [KK13] demonstrate the need for overlapping domains to avoid contour discontinuity at the boundaries. Thus, we increase r, and in overlapping regions, the predicted estimates are averaged among GPs.

Subset of data: Before adding a training point c_i to the active set, we ensure that the output variance σ_i^2 is greater than a pre-defined threshold σ^2_{thresh}. This promotes sparsity in our model by excluding potentially redundant information.

Implementation: Rather than direct matrix inversions (Eq. (2.4)), it is more efficient to use the Cholesky factor of the kernel matrix [Ras03]. In our online setting, we directly update the Cholesky factor $\mathcal{L}\mathcal{L}^T = (K + \sigma^2_{\text{noise}} I)$ with new information. We multi-thread the update/test steps, and perform these at a lower frequency than the graph optimization. We set $L = 25$, grid sampling resolution 5 mm, and circular prior of radius 40 mm for our experiments.

2.4 Pose estimation with factor graphs

2.4.1 Factor graph formulation

The *maximum a posteriori* (MAP) estimation problem gives variables that maximally agree with the sensor measurements. This is commonly depicted as a factor graph: a bipartite graph with variables to be optimized for and factors that act as constraints (Fig. 2.2). The augmented state

comprises object poses and a motion model scalar $\mathcal{C}$:

$$\mathcal{X}_t = \{\mathrm{x}_1, \ldots, \mathrm{x}_t \; ; \; \mathcal{C}\} \tag{2.6}$$

Prior work in pushing empirically validates measurement noise to be well-approximated by a Gaussian distribution [YR18]. With Gaussian noise models, MAP estimation reduces to a non-linear least-squares problem [DK17]. Our MAP solution (given best-estimate shape $\mathcal{S}$) is:

$$\begin{aligned}\mathcal{X}_t^* = \underset{\mathcal{X}_t}{\operatorname{argmin}} \sum_{i=1}^{t} \Big(& \underbrace{||Q(\mathrm{x}_{i-1}, \mathrm{x}_i, \mathrm{z}_{i-1}, \mathcal{C})||^2_{\Sigma_Q}}_{\text{QS pushing factor}} + \underbrace{||I(\mathrm{x}_i, \mathrm{z}_i, \mathcal{S})||^2_{\Sigma_I}}_{\text{IS contact factor}} \\ & + \underbrace{||P(\mathrm{x}_i, \mathrm{z}_i, \mathcal{S})||^2_{\Sigma_P}}_{\text{Non-penetration factor}} + \underbrace{||F(\mathrm{x}_{i-1}, \mathrm{x}_i)||^2_{\Sigma_F}}_{\text{Finite motion factor}} \Big) + \underbrace{||\mathrm{p}_0||^2_{\Sigma_0} + ||\mathrm{c}_0||^2_{\Sigma_c}}_{\text{Priors}}\end{aligned} \tag{2.7}$$

This is graphically represented in Fig. 2.2, and the cost functions are described in Section 2.4.2. Given covariance matrix Σ, $\|v\|^2_\Sigma = v^T \Sigma^{-1} v$ is the Mahalanobis distance of v. The noise terms for covariances $\{\Sigma_Q, \ldots, \Sigma_c\}$ are empirically selected. The online estimation is performed using incremental smoothing and mapping (iSAM2) [Kae+12]. Rather than re-calculating the entire system every timestep, iSAM2 updates the previous matrix factorization with new measurements. In addition, we use a fixed-lag smoother to bound optimization time over the exploration [DK17]. Fixed-lag smoothing maintains a fixed temporal window of states $\mathcal{X}_w$, while efficiently marginalizing out preceding states (100 timesteps in our experiments). Note that this is different from simply culling old states and discarding information.

2.4.2 Cost functions

QS pushing factor: The quasi-static model uses a limit surface (LS) model to map between pusher force $\mathbf{f}_t$ and object motion [GRP91]. Specifically, Lynch et al. [LMT92] develop an analytical model using an ellipsoid LS approximation [LC91]. The factor ensures object pose

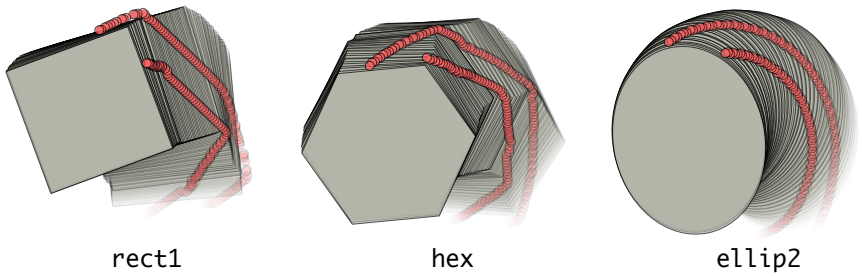


Figure 2.4: Snippets of the tactile exploration data collected in the PyBullet simulator. We use a two-finger pusher to perform contour following, and collect the tactile measurements and ground-truth poses.

transitions obey the quasi-static motion model, with an error term:

$$Q(\mathrm{x}_{t-1},\mathrm{x}_t,\mathrm{z}_{t-1},\mathcal{C}) = \left[\frac{v_x}{\omega} - \mathcal{C}^2\frac{f_{t-1x}}{\tau}, \frac{v_y}{\omega} - \mathcal{C}^2\frac{f_{t-1y}}{\tau}\right] \tag{2.8}$$

- (v_x, v_y, ω) is the object's velocity between x_{t-1} and x_t,
- τ is the applied moment w.r.t. pose center of x_{t-1},
- $\mathcal{C} = \tau_{\text{max}}/f_{\text{max}}$ is an object-specific scalar ratio dependent on pressure distribution.

For a more rigorous treatment, we refer the reader to [YLR15]. We weakly initialize $\mathcal{C}$ with our known circular shape prior, and incorporate it into the optimization.

Implicit surface contact factor: Given $\Theta_t = 1$ (contact), we encourage the measured contact point $\mathbf{c}_t$ to lie on the manifold of the object. We define $\Phi = \Phi(\mathrm{x}_t, \mathrm{z}_t, \mathcal{S})$ that computes the closest point on $\mathcal{S}$ w.r.t. the pusher via projection. The error term is defined as:

$$I(\mathrm{x}_t, \mathrm{z}_t, \mathcal{S}) = \left[\Phi - \mathrm{c}_t \ , \ \|\Phi - \mathrm{p}_t\| - r_{\text{probe}}\right] \tag{2.9}$$

This ensures that in the presence of noise, the contact points lie on the surface and the normals are physically valid [YLR15].

Non-penetration factor: While we assume persistent contact, this cannot be assured when there is more than one pusher. When $\Theta_t = 0$ (no contact) we enforce an intersection penalty (as used in [Lam+19]) on the pusher-slider system. We define $\Psi = \Psi(\mathrm{x}_t, \mathrm{z}_t, \mathcal{S})$ to estimate the pusher point furthest inside the implicit surface, if intersecting. The error is:

$$P(\mathrm{x}_t, \mathrm{z}_t, \mathcal{S}) = \begin{cases} \|\Psi - \Phi\|, & \text{when intersecting} \\ 0, & \text{when not intersecting} \end{cases} \tag{2.10}$$

Finite motion factor: Given persistent contact, we weakly bias the object towards constant motion in $SE\,(2)$. The magnitude is empirically chosen from the planar pushing experiments. We observe this both smooths the trajectories, and prevents an indeterminate optimization.

Priors: The prior p_0 anchors the optimization to the initial pose. $\mathcal{C}$ is initialized with c_0 using the circular shape prior.

2.5 Experimental evaluation

We demonstrate the framework in both simulated (Section 2.5.1) and real-world planar pushing tasks (Section 2.5.2).

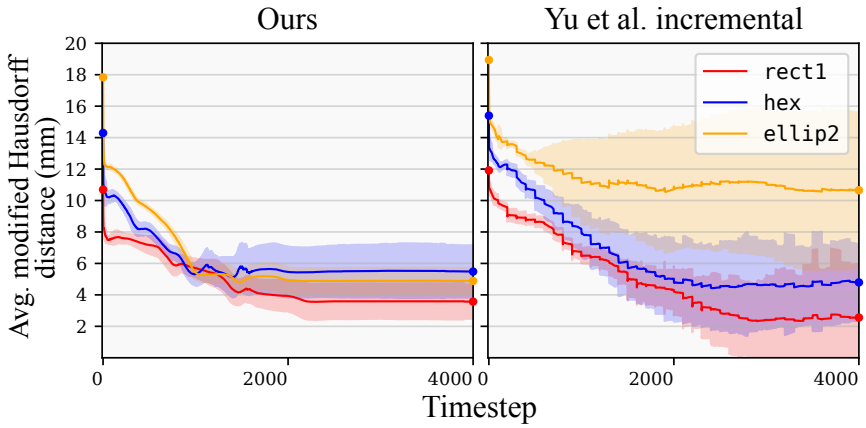


Figure 2.5: Average MHD w.r.t. the ground-truth model for the 50 logs and for each of the 3 objects. Comparing ours **[left]** to *Yu et al. incremental* **[right]**, we see less variance across all shapes, and much lower error in `ellip2`. This shows that while the piecewise-linear representation is suited to polygonal objects, it understandably fails to generalize to more arbitrary shapes. The GPIS faithfully approximates both classes. Moreover, the errors in data association lead to large variance among trials in `rect1` and `hex`.

Evaluation metrics: For pose error, we evaluate the root mean squared error (RMSE) in translation and rotation w.r.t. the true object pose. For shape, we use the modified Hausdorff distance (MHD) [DJ94] w.r.t. the true object model. The Hausdorff distance is a measure of similarity between arbitrary shapes that has been used to benchmark GPIS mapping [SS20], and the MHD is an improved metric more robust to outliers.

Baseline: We compare against the work of Yu et al. [YLR15], which recovers shape and pose as a batch optimization (Section 2.1). For fairness, we implement an online version as our baseline, that we refer to as *Yu et al. incremental*. We use $N_s = 25$ shape nodes in the optimization to better represent non-polygonal objects.

Compute: We use the GTSAM library with iSAM2 [Kae+12] for incremental factor graph optimization. The experiments were carried out on an Intel Core i7-7820HQ CPU, 32GB RAM without GPU parallelization.

2.5.1 Simulation experiments

Setup: The simulation experiments are conducted in PyBullet [Cou10] (Fig. 2.4). We use a two-finger pusher ($r_{\text{probe}} = 6.25$ mm) to perform tactile exploration at 60 mm/s. Contour following is based on previous position and sensed normal reaction. The coefficients of friction of the object-pusher and object-surface are both 0.25. Zero-mean Gaussian noise with std. dev. $[0.1 \text{ mm}, 0.01 \text{ N}]$ are added to $[\mathbf{c}_t, \mathbf{f}_t]$.

We run 50 trials of 4000 timesteps each, on three shape models [Yu+16]: **(i)** `rect1` (90 mm side), **(ii)** `hex` (60.5 mm circumradius), and **(iii)** `ellip2` (130.9 mm maj. axis). While our framework can infer arbitrary geometries, the contour following schema is best suited for simpler planar

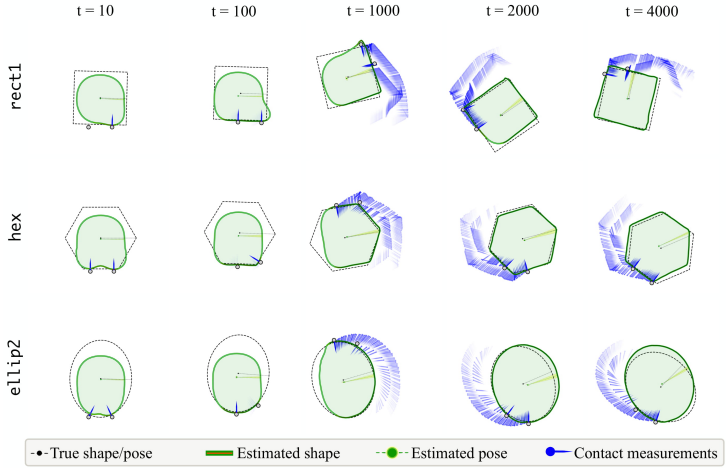


Figure 2.6: Estimated shape and pose of representative simulation trials, with timesteps t. We compare these against the ground-truth, and overlay the stream of tactile measurements.

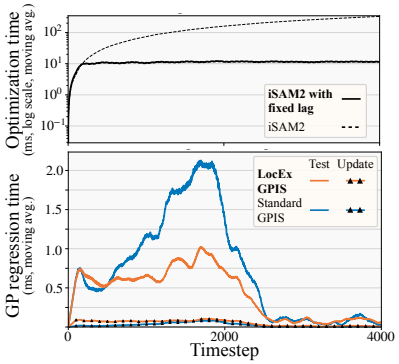


Figure 2.7: Moving average of execution time of the processes, over all 3×50 logs. **[top]** While the complexity of full iSAM2 grows linearly with time, fixed-lag smoothing maintains a bounded optimization window. **[bottom]** As a result of spatial partitioning, local GPIS regression has a lower query time compared to the standard implementation.

objects. The object's initial pose x_0 is randomly perturbed over the range of $\pm(2 \text{ mm}, 2 \text{ mm}, 5°)$.

Results: We highlight the qualitative results of a few representative trials in Fig. 2.6. We observe the evolution of object shape from the initial circular prior to the final shape, with pose estimates that match well with ground-truth.

Fig. 2.5 shows the decreasing MHD shape error over the 50 trials. The uncertainty threshold of the GPIS σ^2_{thresh} (Section 2.3.2) prevents shape updates over repeated exploration, and hence the curve flattens out. This trades-off accuracy for speed, but we find little perceivable difference in the final models. The baseline has larger error for `ellip2`, a shape which their formulation cannot easily represent. Moreover, the uncertainty of the shape estimates are high, due to data association failures. Our representation has no explicit correspondences, and the kernel smooths out outlier measurements. An example of these effects are seen in Fig. 2.13. A similar trend is seen in pose RMSE across all trials (Fig. 2.12). The baseline shows comparable performance only with `rect1`, as the shape control points can easily represent it.

Finally, we quantify the computational impact of both the local GPIS regression and incremental fixed-lag optimizer (Fig. 2.7). Fixed-lag smoothing keeps the optimization time bounded, and prevents linear complexity rise. Spatial partitioning keeps online query time low, and σ^2_{thresh} results in less frequent updates over time for both methods. The combination of these two give us an average compute time of about 10 ms or 100 Hz. For reference, at 60 mm/s, that equates to an end-effector travel distance of 0.6 mm per computation. The maximum time taken by a re-linearization step is 55 ms (3.3 mm travel).

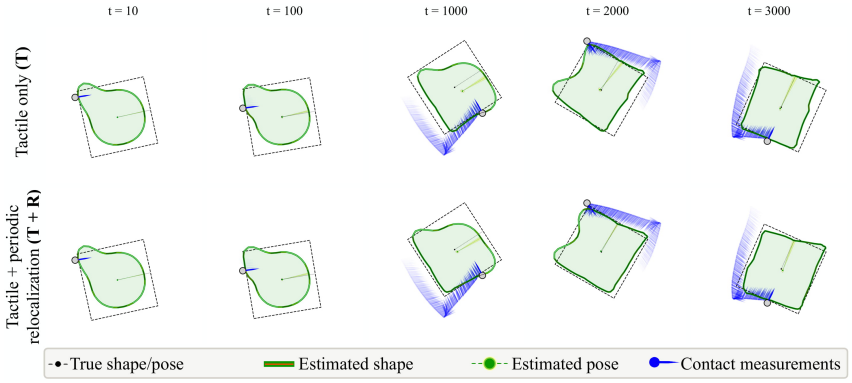


Figure 2.8: Representative results from the real-world estimation task. We compare our tactile only **(T)** result to one aided by periodic relocalization **(T + R)**. We add 10 such events in the trial, and the reduced pose drift improves shape prediction.

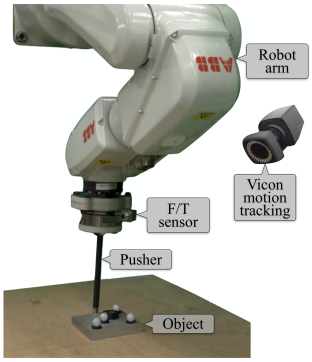


Figure 2.9: Data collection setup: The ABB IRB 120, with F/T sensing, pushes the block on a plywood surface. The Vicon system tracks the object as ground-truth in our evaluation.

2.5.2 Real-world tactile exploration

Setup: We carry out an identical tactile exploration task with the pusher-slider setup in Fig. 2.9. An ABB IRB 120 industrial robotic arm circumnavigates a square object (98 mm side) at the rate of 20 mm/s. We perform the experiments on a plywood surface, with object-pusher and object-surface coefficient of friction both ≈ 0.25. We use a single cylindrical rod with an attached ATI F/T sensor to measure reaction force.

Contact is detected with a force threshold, set conservatively to reduce the effect of measurement noise. Ground-truth is collected with Vicon, tracking reflective markers on the object. We collect three trials of 4000 timesteps each, with tactile measurements and ground-truth. In this case, we do not record force magnitude, but only contact normals. Instead pusher velocity is mapped to forces via the motion cone, and we reason about sticking and slipping [Mas86].

Figure 2.10: RMSE for real-world tactile exploration. Apart from *Yu et al. incremental*, we also compare to a method aided by periodic relocalization.

Method	Trans. RMSE (mm)	Rot. RMSE (rad)
Ours (T)	10.60 ± 2.74	0.09 ± 0.02
Ours (T + R)	4.60 ± 1.00	0.09 ± 0.01
Yu et al. incremental	12.75 ± 4.01	0.17 ± 0.03

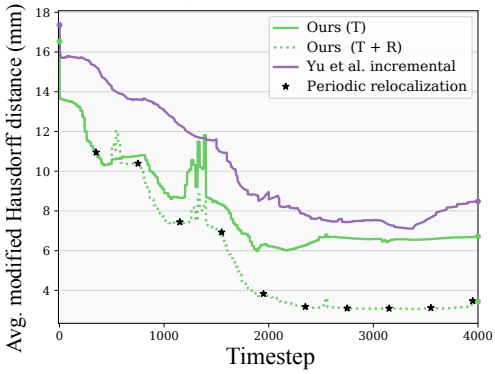


Figure 2.11: Average MHD w.r.t. the ground-truth model for the real-world experiments, compared against the baseline.

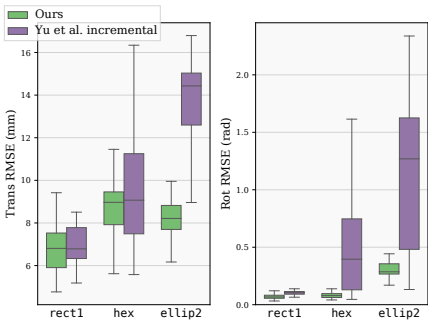


Figure 2.12: Translation and rotation RMSE box-plots across the 50 simulation trials, for each of the objects. *Yu et al. incremental* performs comparably for `rect1`, but has higher variance and error for the more complex shapes.

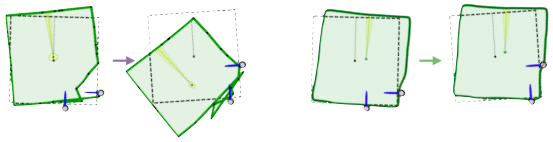


Figure 2.13: **[left]** An example of data association failure in the baseline parametric representation [YLR15]. Without discriminating features, committing to vertex associations affects the entire optimization. **[right]** The GP does not require associations, and the kernel smooths out outlier measurements.

Results: The top row **(T)** of Fig. 2.8 shows the evolution of shape and pose over the runtime. When compared to the simulation results (Fig. 2.6), we notice aberrations in the global shape. We can attribute these to: **(i)** lack of a second pusher, which enables better localization and stable pushes, and **(ii)** motion model uncertainty in real-world pushing.

The bottom row **(T + R)** of Fig. 2.8 describes another scenario, where we demonstrate better results when we periodically relocalize the object. This is a proxy for combining noisy global estimates from vision in a difficult perception task with large occlusions. To illustrate this, we crudely simulate 10 such events over each trial with global Vicon measurements with Gaussian noise. Fig. 2.11 plots the evolution of shape error over the three trials. Decrease in shape error is correlated to relocalization events, highlighting the importance of reducing pose drift. Finally, Fig. 2.10 shows the RMSE of our experiments.

2.6 Discussion and limitations

We formulate a method for estimating shape and pose of a planar object from a stream of tactile measurements. The GPIS reconstructs the object shape, while geometry and physics-based constraints optimize for pose. By alternating between these steps, we show real-time tactile SLAM in both simulated and real-world settings. This method can potentially accommodate tactile arrays and vision, and be extended beyond planar pushing.

In the future, we wish to build on this framework for online SLAM with dense sensors, like the GelSight [YDA17] or GelSlim [Don+18], to reconstruct complex 3-D objects. Multi-hypothesis inference methods [HK19] can relax our assumption of known initial pose, and learned shape priors [Wan+18] can better initialize unknown objects. Knowledge about posterior uncertainty can guide active exploration [DTG13], and perform uncertainty-reducing actions [DS12].

Chapter 3

Visuo-tactile shape reconstruction

ShapeMap 3D

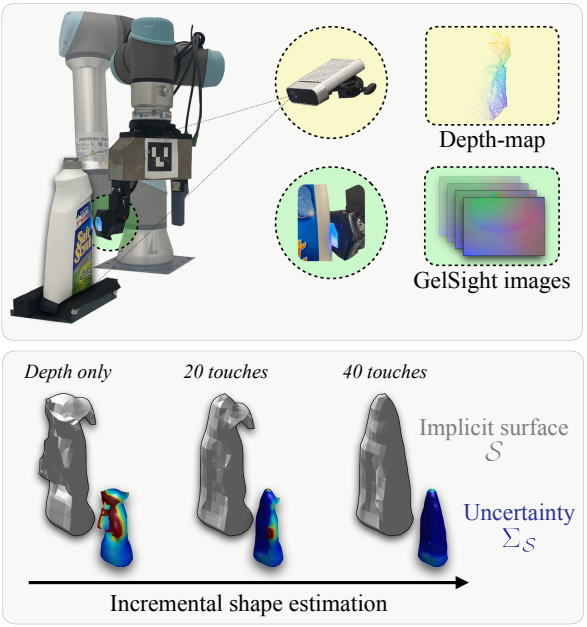


Figure 3.1: We perform incremental 3D shape mapping with a vision-based tactile sensor, GelSight, and an overlooking depth-camera. We combine multi-modal sensor measurements into our Gaussian process spatial graph (GP-SG), for efficient incremental mapping. The depth-camera gives us an occluded noisy estimate of 3D shape, after which we sequentially add tactile measurements as Gaussian potentials into our GP-SG. The tactile measurements are recovered from GelSight images via a learned model trained in simulation. The results demonstrate accurate implicit surface reconstruction and uncertainty prediction for interactive perception tasks.

Vision-based touch has higher spatial acuity than point-contact or tactile arrays, which lends itself to 3D reconstruction [Wan+18; BCR19; Smi+20; Smi+21]. A key challenge is to efficiently incorporating these dense measurements into a 3D mapping framework. Moreover, the tactile sensor's coverage is limited by its size and durability, and cameras only provide partial visibility of the object. It's desired that a shape representation can faithfully approximate regions lacking measurements.

This chapter is adapted from the publication *ShapeMap 3-D: Efficient shape mapping through dense touch and vision* by Sudharshan Suresh, Zilin Si, Joshua Mangelson, Wenzhen Yuan, and Michael Kaess. In Proc. IEEE Intl. Conf. on Robotics and Automation, ICRA, (Philadelphia, USA), May 2022.

Studies show humans can optimally fuse touch and vision to reconstruct shape [HE07], reinforcing their complementarity. Vision gives coarse global context, while touch gives precise local information. The development of vision-based tactile sensing [YA16; YDA17; Don+18; War+18; Als+19; Lam+20; Pad+20; Wan+21], like the GelSight [YDA17], has led to renewed interest in the shape mapping problem. Fusing both modalities requires globally integrating tactile signals at the distal end, joint kinematics, and vision.

We propose ShapeMap 3D, a framework that incrementally reconstructs 3D objects from a sequence of tactile images and a noisy depth-map (Fig. 3.1). We leverage optical tactile simulation to learn local shape from GelSight-object interactions. We represent 3D shape as a signed distance function (SDF) sampled from a Gaussian process (GP), and re-formulate shape mapping as probabilistic inference on a spatial graph. We show that visuo-tactile measurements can be incorporated into a factor graph as local Gaussian potentials. This affords efficient access to the implicit surface and SDF uncertainty. We present both simulated and real experiments, generating reconstructions of global shape with trajectories of limited sensor coverage. Specifically, our contributions include:

(1) Recovery of local shape from touch learned via simulation of GelSight-object interactions,
(2) Incremental shape mapping through our Gaussian process spatial graph (GP-SG),
(3) Task evaluation on our *YCBSight-Sim* and *YCBSight-Real* datasets.

3.1 Related work

Tactile sensing and local shape

For vision-based tactile sensors, photometric stereo [HS05] has been widely used to reconstruct local shape [JA09; Joh+11; YDA17]. The approach maps image intensities to gradients via a lookup table, and integrates the gradients to obtain a height-map. However, this method does not consider spatial position in the calibration, and leads to large variance around the boundary of the sensor. Later works learn these gradients directly from tactile images, via either a multilayer perceptron [Wan+21] or pix2pix networks [Sod+22b]. Alternatively, end-to-end learning [BCR19; Amb+21] from a limited set of real-world tactile interactions can directly provide heightmaps. Tactile simulation [Bau+20; AMY21; Wan+22; SY22] allows us to scale supervised-learning to a wide range of objects and ground-truth.

Visuo-tactile shape perception

Global information from vision has complemented low-resolution touch in a multi-modal setting [Bjö+13; IBK14; Var+17; Gan+20]. Wang et al. [Wan+18] use monocular shape completion augmented with GelSight readings. However they rely primarily on the visual shape prediction, and tactile sensing serves as a refinement step. Smith et al. [Smi+20; Smi+21] demonstrate a learned

perception model on simulated datasets, to predict local mesh deformations via high-resolution touch and filling-in through vision. The context of our work resembles those of [Wan+18] and [Smi+20], with partial vision and high-dimensional touch. Our contributions differ from these methods as we **(i)** perform incremental inference on the measurement stream, and **(ii)** do not rely on data-driven shape priors.

Gaussian processes and graphs

We wish to faithfully approximate non-contact regions, capture surface uncertainty, and probabilistically handle measurement noise. Gaussian process implicit surfaces (GPIS) [WF07] showcase these properties and have found preference in manipulation research—over point-clouds [BCR19] and other parametric methods [BGD08]. The GPIS considers the object's SDF magnitude and gradient as a GP, conditioned on noisy sensor measurements. This has been successfully applied to both passive [DTG11; Ott+16] and active 3D reconstruction [Bjö+13; Jam+16; Yi+16; DET17] with low-resolution tactile data. We extend these ideas, scaling them to a stream of high-dimensional touch measurements for incremental shape reconstruction.

The key challenge, especially for GelSight point-clouds, is that GPs scale poorly due to matrix inversion costs. In the SLAM community, common approximations include local GPs [Lee+19; SS20] and compact kernels [RYH10]. These have further been incorporated into factor graphs [DK17] for trajectory estimation [YIB17], target tracking [RHL14], motion planning [Muk+18], elevation modeling [WSE19], and planar mapping [Sur+21]. Inspired by these, our representation encodes GP potentials as local constraints in a spatial factor graph.

3.2 Problem formulation

We consider a robot arm with a GelSight tactile sensor interacting with an unknown 3D object fixed on a tabletop. Given a sequence of images from the GelSight, robot kinematics, and depth-map from a depth-camera, we incrementally estimate the object's shape and signed distance function (SDF) uncertainty.

Object shape: We represent the object's shape as an implicit surface $\mathcal{S} \in \mathbb{R}^3$ in the robot's frame, with SDF uncertainty $_{\mathcal{S}}$ (Refer Section 3.4).

Tactile measurements: During interaction, upon detecting contact, we record the corresponding tactile image $\mathrm{I_t}$ and sensor pose $\mathrm{p_t}$:

$$\mathrm{z}_t = \left\{ \mathrm{I}_t \in \mathbb{R}^{640\times480\times3},\ \mathrm{p}_t \in SE\,(3) \right\} \tag{3.1}$$

Depth-map: We capture a depth-map D_0 of the object from the camera, represented in the robot-frame: $\mathrm{d}_{1\cdots M} \in \mathbb{R}^3$.

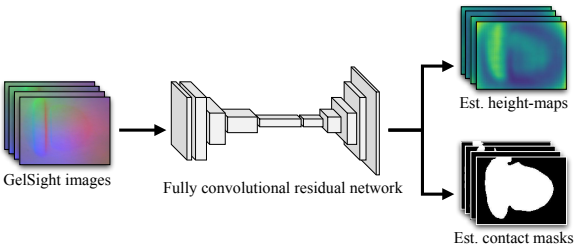


Figure 3.2: The depth-network takes in tactile images, and outputs both estimated height-maps and binary contact masks. We train on a corpus of GelSight-object interactions in simulation.

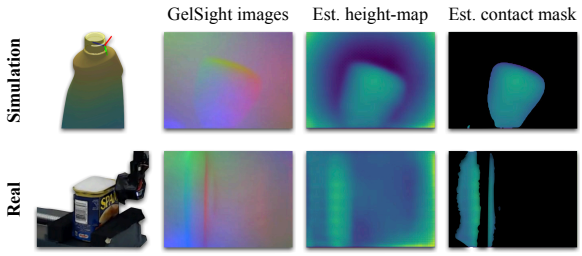


Figure 3.3: Tactile images generated from GelSight interactions in **[top]** simulated and **[bottom]** real settings. Pictured alongside are the height-maps and contact masks output from our learned model Ω.

Assumptions: In line with prior efforts, we assume:

- Calibrated robot-camera extrinsics,
- Fixed object pose and known approximate object dimensions,
- A passive exploration algorithm for object coverage.

3.3 Local shape from touch

Vision-based tactile sensors perceive contact geometries as images. The soft, illuminated gelpad deforms elastically on contact and is captured by an embedded camera. We represent local shape recovery as the inverse sensor model:

$$\Omega : \mathrm{I}_t \mapsto \mathrm{H}_t, \mathrm{C}_t, \text{ where } \begin{matrix} \mathrm{I}_t\text{: tactile image, } \mathrm{H}_t\text{: recovered height-map,} \\ \mathrm{C}_t\text{: contact area mask} \end{matrix} \tag{3.2}$$

In this section, we learn Ω through simulation, and its output forms the basis for our visuo-tactile mapping in Section 3.4. With H_t, C_t, and knowledge of sensor pose p_t from robot kinematics, we can obtain a tactile point-cloud $\boldsymbol{\mathcal{M}}_t$, comprising of 3D points m_i^x and normals m_i^y:

$$\boldsymbol{\mathcal{M}}_t = \{[\mathrm{m}_1^x, \mathrm{m}_1^y], [\mathrm{m}_2^x, \mathrm{m}_2^y], \cdots\} \tag{3.3}$$

For tactile sensors with soft body deformation, local shape geometry can be learned through supervision. Image-to-depth estimation networks [EPF14; Lai+16] can learn accurate heightmaps from GelSight images. This would require a large corpus of tactile images and corresponding ground-truth depths, for which we can leverage tactile simulation. In particular, Si et al. [SY22] calibrate their simulator with real-world tactile images, thus mimicking the same intensity distributions.

Network and training: We use an implementation [XPF18] of the fully convolutional residual network [Lai+16] as our depth estimator, as shown in Fig. 3.2. The network combines ResNet-50

as the encoder and up-sampling blocks as the decoder. Our model takes tactile images as input, and outputs predictions of both height-map H_t and contact mask C_t. We choose 30 household objects from YCB dataset [Cal+17], and hold out 6 objects for testing generalization. For each object, we generate 660 images from randomly sampled sensor poses on their ground-truth mesh models. We split the train-validation-test sets as 550-50-60.

Benchmarks: We compare Ω with the standard lookup table method [YDA17]. This maps tactile images to gradients of the local shape, and fast Poisson integration generates their height-maps. The lookup is built via a calibration routine with a 4 mm sphere indenter. The contact masks are generated by intensity thresholding of contact vs. non-contact frames.

Evaluation: Fig. 3.4 compares Ω with respect to benchmarks on our *YCBSight-Sim* dataset (refer Section 3.5.1). We compare each estimated height-map and contact mask against the ground-truth. Specifically, we evaluate:

(i) Pixel-wise RMSE on height-maps, and

(ii) Intersection over union (IoU) on contact masks.

On height-map estimation, we outperform the benchmark with an average RMSE of 0.094 mm across all object classes. The lookup table has larger variance, with an average RMSE of 0.182 mm. Note that the maximum penetration depth of the simulation is 1 mm. On contact mask estimation, we have an average IoU of 0.752, while the handcrafted image thresholding performs worse with 0.379. In addition, the IoU variance appears to be larger for objects with more intricate shapes. Finally, in Fig. 3.3, we show generalization of Ω to both unseen simulation and real-world tactile interactions.

3.4 3D shape estimation

Standard Gaussian processes

A GP is a nonparametric method to learn a continuous function from data, well-suited to model spatial and temporal phenomena [Ras03]. To estimate shape, a GP considers the object's SDF to be a joint Gaussian distribution over noisy measurements of its surface. At any spatial position, the SDF φ represents the signed-distance from the surface: $\varphi = 0$ on the surface, $\varphi < 0$ inside, and $\varphi > 0$ outside. The GP meaningfully approximates the global shape, even in regions lacking sensor information. Given tactile measurement $\mathrm{m}_i \in \boldsymbol{\mathcal{M}}_t$[2], we learn a function between positions m_i^x and normals m_i^y:

$$\mathrm{m}_i^x \mapsto [\varphi = 0,\ \mathrm{m}_i^y] \tag{3.4}$$

[2]or depth map $\mathrm{d}_{1\cdots M} \in \mathrm{D}_0$

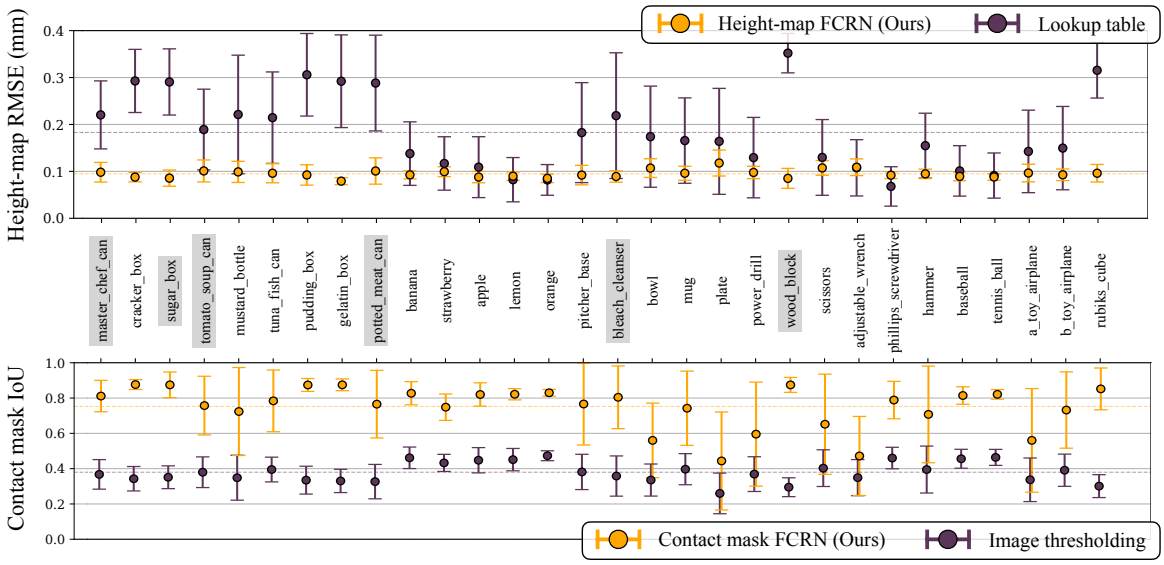


Figure 3.4: Local shape recovery benchmarked on our *YCBSight-Sim* dataset (Refer Section 3.5.1). **[top]** We evaluate our learned model with respect to the baseline lookup table method for height-map estimation. Here we use a pixel-wise root-mean-square error (RMSE)↓ metric, and observe consistent, low error for our method when compared with the lookup table. **[bottom]** We compare our learned contact mask model against intensity-based thresholding on interover union (IoU)↑ metric. The test data is randomly generated, and gray represents the hold-out objects not encountered in training.

More generally, treating the left and right hand side of Eq. (3.4) as the input-output:

$$\mathrm{X} = \{\mathrm{x}_i \in \mathbb{R}^3\}^{1\cdots N} \mapsto \mathrm{Y} = \{\mathrm{y}_i \in \mathbb{R}^4\}^{1\cdots N} \tag{3.5}$$

The posterior distribution at a query point $(\mathrm{x}_j^*, \mathrm{y}_j^*)$ for a full GP with N measurements, is given by [Ras03]:

$$\mathrm{y}_j^* \sim \mathcal{GP}\big(\underbrace{k_*^T \left(K + \sigma_\mathrm{n}^2 I\right)^{-1} Y}_{\text{mean}},\ \underbrace{k_{**} - k_*^T \left(K + \sigma_\mathrm{n}^2 I\right)^{-1} k_*}_{\text{variance}}\big) \tag{3.6}$$

where σ_n is the sensor noise covariance, and $K \in \mathbb{R}^{4N\times 4N}$, $k_* \in \mathbb{R}^{4N\times 4}$ and $k_{**} \in \mathbb{R}^{4\times 4}$ are the train-train, train-query, and query-query kernels respectively. Each kernel's constituent block $k_{ij} = k(x_i,\ x_j)$ is an $\mathbb{R}^{4\times 4}$ kernel basis, in our case a thin-plate function [WF07]. This inference is computationally intractable for the large N that accrues from high-dimensional tactile measurements. The update operations involve costly $O(N^3)$ matrix inversions, and per-query costs $O(N^2)$ (Refer Eq. (3.6)). We now present a local approximation that can be updated and queried incrementally, with bounded computational costs.

GP-SG: Gaussian process spatial graph

We represent the scene as a spatial factor graph [DK17], comprising of nodes we optimize for and factors that constrain them. These query nodes Y^* are at their respective spatial positions X^*, distributed in an S^3 volume. Our optimization goal is to recover the posterior $\hat{\mathrm{Y}}^*$, which

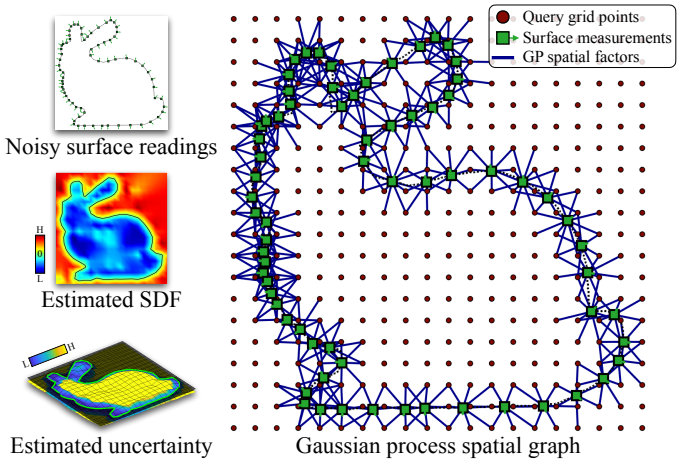


Figure 3.5: [right] A 2-D illustration of our GP spatial graph (GP-SG), an efficient local approximation to a full GP. The graph consists of SDF query nodes (●) Y^* each at their spatial positions X^*. Each surface measurement (■) $(\mathrm{x}_i, \mathrm{y}_i)$ produces a unary factor (–) $\mathcal{G}_{ij}$ at query node y_j^* (within the local radius r). This represents a local Gaussian potential for the GP implicit surface. **[left]** Optimizing for $\hat{\mathrm{Y}}^*$ yields posterior SDF mean + uncertainty. The zero-level set of the SDF gives us the implicit surface $\mathcal{S}$.

represents the SDF of the volume and its underlying uncertainty. Implementing the full GP (Eq. (3.6)) in the graph is costly, as each measurement $(\mathrm{x}_i, \mathrm{y}_i)$ constrains all query nodes Y^*. Motivated by prior work in spatial partitioning [Lee+19; SS20], we decompose the GP into local unary factors as a sparse approximation. Given that y_i and query node y_j^* follow a GP, the joint distribution and conditional are:

$$\begin{aligned} \begin{bmatrix} \mathrm{y}_i \\ \mathrm{y}_j^* \end{bmatrix} &\sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} k_{ii} + \sigma_{\mathrm{n}}^2 I & k_{i*} \\ k_{*i} & k_{**} \end{bmatrix} \right) \\ \mathrm{y}_j^* \mid \mathrm{y}_i &\sim \mathcal{N}\Big(\underbrace{k_{*i}\left(k_{ii} + \sigma_{\mathrm{n}}^2 I\right)^{-1} \mathrm{y}_i}_{\mu_{\mathrm{y}_j^* \mid \mathrm{y}_i}}, \underbrace{k_{**} - k_{*i}^2 k_{ii}^{-1}}_{\Sigma_{\mathrm{y}_j^* \mid \mathrm{y}_i}} \Big) \end{aligned} \tag{3.7}$$

This gives us a unary Gaussian potential which can fit into a least-square setting:

$$\mathcal{G}_{ij} = ||\mathrm{y}_j^* - \mu_{\mathrm{y}_j^* \mid \mathrm{y}_i}||^2_{\Sigma_{\mathrm{y}_j^* \mid \mathrm{y}_i}} \tag{3.8}$$

At a timestep t, given n measurements $(\mathrm{x}_i, \mathrm{y}_i)$, we add the set of associated factors within a local radius r of each query node's position x_j^*. r represents a tradeoff between speed and reconstruction accuracy, and is empirically set to 15% of the ground-truth object's side length. Thus, for all query nodes, we accumulate a small set of factors:

$$\mathcal{R}_t \triangleq \left\{ \{\mathcal{G}_{ij}\}_{\substack{i=1\cdots n \\ j=1\cdots S^3}} \;\middle|\; \|\mathrm{x}_j^* - \mathrm{x}_i\| \leq r \right\} \tag{3.9}$$

This sparsifies an otherwise intractable optimization, pictorially represented in Fig. 3.5 for a 2-D case. Taking the Stanford bunny as an example, we illustrate how a set of noisy surface measurements are converted into local GP factors. The final optimization recovers a posterior SDF mean and uncertainty. More specifically, for the visuo-tactile problem, the *maximum a posteriori* estimation is:

$$\hat{\mathrm{Y}}^{*} = \underset{\mathrm{Y}^{*}}{\operatorname{argmin}} \overbrace{\sum_{\mathcal{G}_d \in \mathcal{R}_0} \mathcal{G}_d}^{\text{depth factors}} + \overbrace{\sum_{t=1}^{T} \sum_{\mathcal{G}_m \in \mathcal{R}_t} \mathcal{G}_m}^{\text{tactile factors}} + \overbrace{\sum_{\mathrm{y}^{*} \in \mathrm{Y}^{*}} ||\mathrm{y}^{*} - \mathrm{b}||^2_{\Sigma_\mathrm{b}}}^{\text{GP priors}} \tag{3.10}$$

where $\mathcal{R}_0$ is the factor set from the depth-map D_0, and $\mathcal{R}_t$ is the factor set from tactile measurement $\boldsymbol{\mathcal{M}}_t$. The term b applies a positive SDF prior to nodes, initializing the volume as empty space. Inference is carried out at each timestep via incremental smoothing and mapping (iSAM2) [Kae+12]. This framework combines the computational benefits of an online local GPIS [Lee+19; SS20] with those of an incremental least-squares solver. This is well-suited for sensors like the GelSight, as the dense point-clouds are too expensive to incorporate into a full GP. When querying, we recover the posterior mean and covariance only for the nodes updated—the remaining grid is accessed from cache.

Implicit surface generation

The posterior estimate $\hat{\mathrm{Y}}^{*}$ represents the SDF's mean and uncertainty, sampled from the S^3 volume. A marching cubes algorithm [LC87] can give us both the implicit surface $\mathcal{S}$ and the corresponding SDF uncertainty $\Sigma_{\mathcal{S}}$. $\mathcal{S}$ is generated as the zero-level set of the SDF:

$$\mathcal{S} \triangleq \{\mathrm{s} \in \mathbb{R}^3 \mid \widehat{\mathrm{Y}}^{*}(\mathrm{s})_{\varphi} = 0\} \tag{3.11}$$

Finally, we prune faces/vertices from $\mathcal{S}$ that lie outside r for *any* of the sensor measurements. These areas have high surface uncertainty, and our spatial graph will poorly approximate them. Furthermore, this is necessary for sequential data as we cannot expect a watertight mesh from partial coverage.

3.5 Experimental evaluation

We illustrate our method in simulated (Section 3.5.2) and real-world (Section 3.5.3) visuo-tactile experiments. The shape estimates are compared with respect to the ground-truth using the Chamfer distance (CD) [Bar+77], a commonly-used shape similarity metric.

Implementation: The framework is executed on an Intel Core i7-7820HQ CPU, 32GB RAM without GPU parallelization. We use the GTSAM [Del12] optimizer with iSAM2 [Kae+12] for incremental inference. Due to the precision of sensing, we empirically weight the noise of tactile measurements lower than that of the depth-map. We set the grid size $S = 16$, which can be increased for higher-resolution reconstructions.

3.5.1 Visuo-tactile data collection

We collect the *YCBSight-Sim* and *YCBSight-Real* datasets for evaluating our method. This comprises of YCB ground-truth meshes [Cal+17], GelSight images from interaction, sensor poses, and a depth-map. While we consider 30 household objects in simulation, we restrict our shape

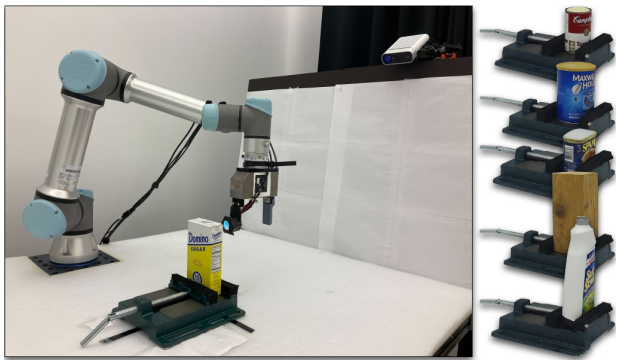


Figure 3.6: Experimental setup for the *YCBSight-Real* dataset, with a GelSight tactile sensor, a depth-camera and the YCB objects. Objects are firmly secured on a mechanical bench vise, to ensure they stay stationary. We collect measurements by approaching from a discretized set of angles and heights, and detecting contact from the tactile images. The overlooking Kinect collects a depth-map to initialize our visuo-tactile mapping.

mapping evaluation to 6 objects. This subset of objects have varied geometries (curved, rectangular, and complex) to verify the generalization of our method.

YCBSight-Sim: We generate GelSight-object interactions using *Taxim*, an example-based tactile simulator [SY22]. We simulate 60 uniformly spread sensor poses on each object, normal to the local surface of the mesh. We render a depth-map from the perspective of an overlooking camera using Pyrender [Mat19]. Finally, zero-mean Gaussian noise is added to tactile point-clouds, sensor poses, and depth-map.

YCBSight-Real: We use a UR5e 6-DoF robot arm, mounting the GelSight sensor on a WSG50 parallel gripper. The depth-map is captured via a fixed-pose, calibrated Azure Kinect, approximately 1m away from the object. Our complete setup can be seen in Fig. 3.6. The GelSight captures 640×480 RGB images of the interactions in a 2.66 cm^2 area. The objects are secured by a mechanical bench vise at a known pose, to ensure they remain static. After capturing the depth-map D_0, we approach each object from a discretized set of angles and heights. This simple strategy works well for the selected objects, and future work can replace this with a closed-loop planner. We detect contact events by thresholding the tactile images. We collect 40 tactile images $\{\mathrm{I}_1, \ldots, \mathrm{I}_{40}\}$ of the object's surface ($\approx$ 20 minutes) with the corresponding gripper poses $\{\mathrm{p}_1, \ldots, \mathrm{p}_{40}\}$ via robot kinematics.

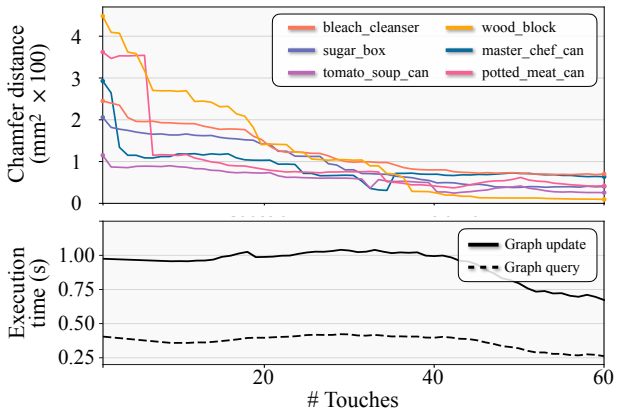


Figure 3.9: **[top]** The Chamfer distance (CD) with respect to ground-truth meshes for *YCBSight-Sim* experiments. Objects are initialized with high CD from partial depth-map, but converge to low-error in 35–40 touches. **[bottom]** Average execution time for update/query operations on our GP spatial graph (GP-SG). At each touch we add $\approx 10^3$ GP factors during update, and recover posterior mean/covariance during query. We see a dip in timing towards the end, due to smaller contact areas on the top of objects.

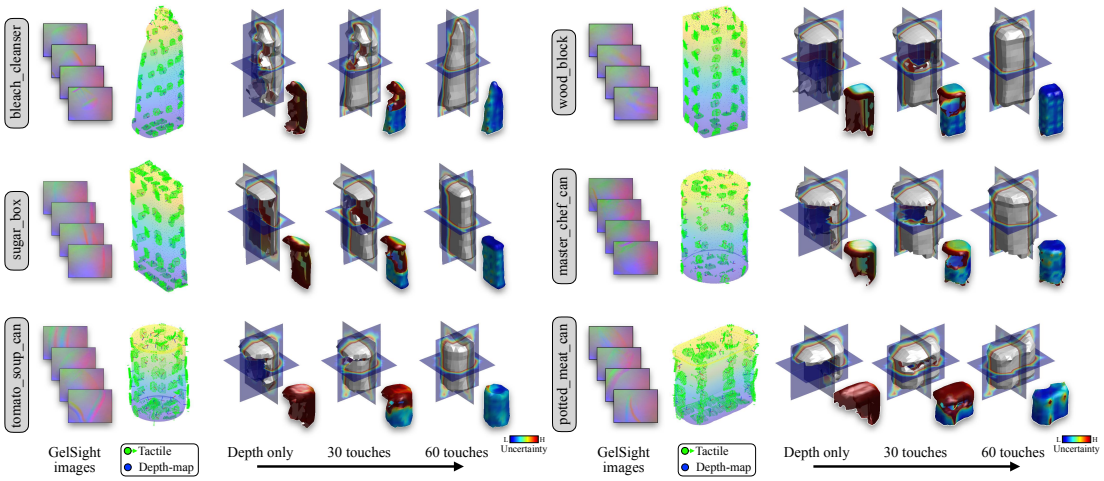


Figure 3.7: Results from simulated visuo-tactile mapping on our *YCBSight-Sim* dataset. Shown for each object are (i) sample GelSight images, (ii) tactile and depth-map measurements on the ground-truth mesh, and (iii) frames from the incremental mapping. Each object is initialized with a noisy rendered depth-map (*Depth only*), and with each sequential GelSight measurement, we gain further understanding of global shape and reduce surface uncertainty. Visualized here are the implicit surface + SDF + uncertainty for the intervals of $[0, 30, 60]$ touches.

3.5.2 Simulated tactile mapping

In Fig. 3.7 we highlight mapping results for the 6 objects in *YCBSight-Sim*. We first visualize the implicit surface and SDF uncertainty from depth-map only. After this, touch measurements are added incrementally and reflect in the shape estimate. The surface uncertainty is typically high for regions that lack depth/tactile information, and reduces over time. Fig. 3.9 shows that the CD with respect to the ground-truth mesh decreases with greater number of touches, and converges within 35–40 touches. The timing plot of graph operations shows near-constant graph update and query time. This reduces towards the end of the datasets due to smaller contact areas on the top surface of the objects. These timings can be further improved by parallelizing spatial operations.

3.5.3 Real-world tactile mapping

In Fig. 3.8, we show our method working on real data collected in *YCBSight-Real*. The Kinect depth-maps for specular objects like `tomato_soup_can` and `potted_meat_can` are erroneous, but tactile information provides more precise local shape. To prevent damage to the robot and sensor, we do not explore near the base of the object—we instead hallucinate measurements at the bottom based on the nearest corresponding sensor poses. In Fig. 3.10, we plot the CD over time for the 6 YCB objects. The initial error is lower than simulation due to the additional

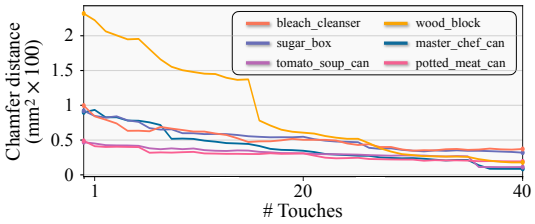


Figure 3.10: The Chamfer distance (CD) with respect to ground-truth meshes for our *YCBSight-Real* experiments. We observe the error converges to a similar magnitude as Fig. 3.9 after 30 touches. They initially start out with a lower error than simulation as a result of the hallucinated base measurements we add to each object (refer Section 3.5.3).

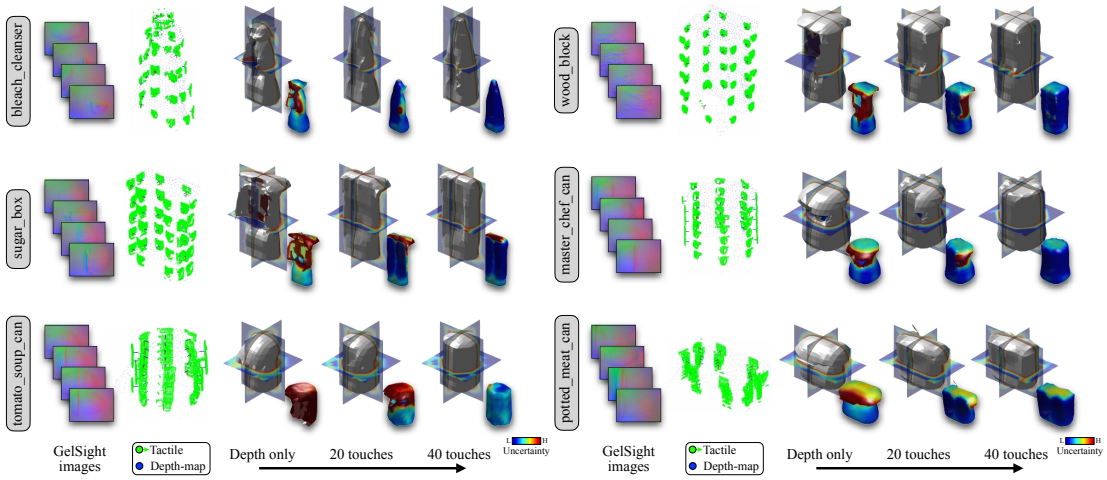


Figure 3.8: Results from real visuo-tactile mapping on our *YCBSight-Real* dataset. This is structured similar to Fig. 3.7, except with reconstruction frames at intervals of $[0, 20, 40]$ touches. The Kinect performs poorly for specular objects such as `tomato_soup_can` and `potted_meat_can`, but high-precision GelSight measurements can disambiguate global shape. Our mapping generalizes well and we observe similar results between simulated and real experiments.

hallucinated measurements. We see the error converge to an average CD of 18.3 mm^2, a similar magnitude as in the simulated experiments. For reference the average diagonal of the ground-truth YCB objects is 20 cm.

3.6 Discussion and limitations

We present an incremental framework for 3D shape estimation from dense touch and vision. We formulate a GP spatial graph (GP-SG) structure, that efficiently infers an object's implicit surface and SDF uncertainty. To integrate GelSight tactile images, we recover local shape with a model learned in tactile simulation. Our method is first demonstrated in a simulated visuo-tactile setting, and is later shown to generalize to real-world shape perception.

As future work, we wish to actively reconstruct these shapes using surface uncertainty information. The current method can further benefit from (i) parallelized spatial graph operations, and (ii) data-driven shape priors [Var+17; Wan+18]. Finally, we wish to consider relaxing the fixed-pose assumption [Sur+21], and perception of deformable objects.

Chapter 4

Global localization from touch

MidasTouch

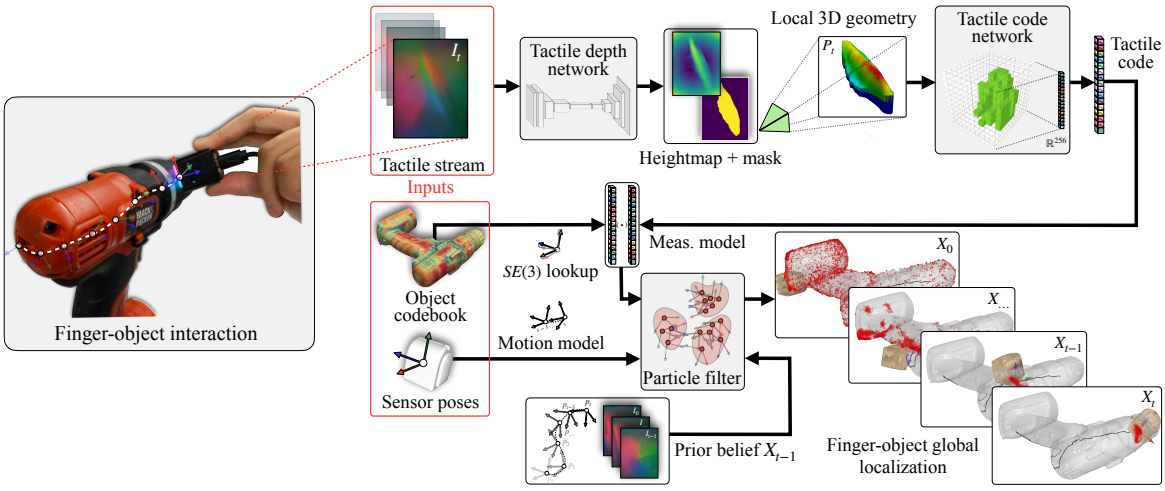


Figure 4.1: MidasTouch performs online global localization of a vision-based touch sensor on an object surface during sliding interactions. Given posed tactile images over time, this system leverages local surface geometry within a nonparametric particle filter to generate an evolving distribution of sensor pose on the object's surface.

Interactive perception is the Catch-22 [Hel61] of robotics; vision can be used to track objects for downstream contact-rich interactions, but such interactions can occlude visual tracking. In particular, end-effector-object relative positioning is crucial for contact-rich policies like sliding [Shi+17; CHR18; She+21; Ker+22b], in-hand manoeuvres [Ope+18; CXA22], finger-gaiting [SH18], and multi-finger enclosure [Bra+19]. Additionally, vision is often affected by object transparency, specularity, and poor scene illumination.

This chapter is adapted from the publication *MidasTouch: Monte-Carlo inference over distributions across sliding touch* by Sudharshan Suresh, Zilin Si, Stuart Anderson, Michael Kaess, and Mustafa Mukadam. In Proc. Conf. on Robot Learning, CoRL, (Auckland, NZ), Dec. 2022

With high-dimensional vision-based tactile sensors [YDA17; Don+18; Lam+20], we now have a window into local object interactions. While tactile images from these sensors capture local surface geometries, they lack global context necessary for relative pose tracking. We address this challenge with *MidasTouch*, an online tactile perception system that tracks the evolving pose distribution of a vision-based touch sensor sliding across known objects. We acquire global context by integrating local interaction observations over long horizons.

The form-factor of vision-based tactile sensors has restricted prior methods to small parts [Li+14; Bau+20; BBR22] or local tracking [Sod+21; Sod+22b]. For everyday objects, we can leverage compositionality [Haw+19]; each is made up of local geometries that are spatially fixed relative to one another. Consider a simple mug: made up of a curved body, flat base, rounded handle, and sharp lip. Without global context, a single-touch is ambiguous: a perceived sharp edge could lie anywhere along the lip of the mug. Such a likelihood distribution is spread across the object's surface and not unimodal, but interaction over long time horizons can disambiguate it. This mirrors haptic *apprehension*, or the exploratory procedures humans perform when presented with a familiar object [LK87].

We approach this as an analogue to mobile robot Monte-Carlo filtering [Del+99], but instead apply it on the surface manifold of the object. Just as a mobile robot has access to detailed floor plans, odometry, and cameras, manipulators have access to object meshes, end-effector poses, and vision-based touch. This is applicable to environments with known object models like households, warehouses, and factories, further facilitated by large-scale scanned object datasets [Cal+17; Dow+22]. While priors from vision can bootstrap our system [Den+21], they are not a prerequisite for global estimation.

In addition to open-sourcing MidasTouch, we will further release a comprehensive real-world and simulated dataset of sliding DIGIT [Lam+20] interactions across standard YCB objects [Cal+17] with ground-truth. This serves as both an evaluation of MidasTouch, and as a benchmark currently lacking in the tactile sensing community. Specifically, our contributions are:

1. Online particle filtering over posed tactile images for a distribution of finger-object poses,
2. Learned embeddings for vision-based touch using local surface geometry,
3. The *YCB-Slide* dataset of object interactions for evaluation and benchmarking.

Our framework relies on recent developments in tactile sensing, rendering, and robotics: **(i)** vision-based tactile sensors [YA16; YDA17; Don+18; War+18; Als+19; Lam+20; Pad+20; Wan+21] like the GelSight and DIGIT have the requisite spatial acuity to discern local geometric features, **(ii)** tactile simulation [Wan+22; AMY21; SY22] with realistic rendering enables learning tactile observation models and precomputing object-specific interactions with sizeable data, which would be infeasible in the real-world, and **(iii)** learned models for 3D place recognition [CPK19; UL18; Kom21] spawned by ubiquitous LIDAR and RGB-D data that can be extended to tactile sensing.

4.1 Related work

Tactile pose inference

Binary contact sensors have been used in conjunction with particle filters for global estimation of robot hands relative to simple geometries [CP10; PK11; ZT12; CRP13; SCS17; Kov+17]. These methods require a large amount of touches, but serve as a touchstone for Monte-Carlo methods that have seen great success in mobile robotics [Del+99; Thr02a]. With tactile arrays, local patch measurements are integrated for planar estimation [PRH11; Luo+15] or for 3D alignment [Bim+16]. The synergy of vision and touch gives much needed global context [YR18; Cha+22] but is outside our current work's scope.

With vision-based touch, Li et al. [Li+14] show planar small-part localization by directly computing the homography on GelSight heightmaps. Relative pose-tracking has also been learnt directly from tactile images either via recurrent [DR19], or auto-encoder [Lam+20; Sod+21] networks. Additionally, local tracking has been explored with online factor-graph optimization [Sod+21; Sod+22b; KR22]. The aforementioned approaches are unimodal and rely on good pose initializations; MidasTouch is nonparametric and requires no such knowledge. Additionally, Kelestemur et al. [KPP22] show category-level object pose estimation with a parallel jaw gripper.

Closely related is Tac2Pose [BBR22; Bau+20], which performs pose estimation of small-parts with the GelSlim [Don+18]. It produces a distribution of object poses learned in tactile simulation from contact shapes. Similarly, Gao et al. [Gao+22] estimates contact location on objects through vision-based touch and audio with a small, discrete set of measurements. We differentiate our work in both context and approach: **(i)** we filter over long time horizons rather than single/multi touch predictions, **(ii)** tactile embeddings are learned from local surface geometry rather than images, and **(iii)** we consider objects considerably larger than the robot finger.

Place recognition for touch

A compact representation for tactile images enables easy frame-to-frame or frame-to-model tracking. Inspired by the computer vision community, a popular intermediate is a learned embedding from either RGB [Lam+20; Sod+21] or binary contact masks [Bau+20; BBR22]. Realistic tactile simulators, like TACTO [Wan+22], enable training these models to the scale and generality of arbitrary real-world interactions. For example, Bauza et al. [BBR22] learn object-specific embeddings to match 2D contact shapes.

For contact-rich manipulation, local 3D geometry is a natural candidate, and iterative closest point (ICP) has shown promise for frame-to-frame tracking [Kup+19; Sod+22b]. For vision-based touch, 3D geometry is obtained either via photometric stereo [JA09; Joh+11; YDA17], or image-to-heightmap models [BCR19; Wan+21; Amb+21; Sod+22b; Sur+22b]. However, ICP is only suitable for local tracking and not global localization: it is sensitive to initialization and is

intractable to scale as a sampling-based measurement model.

Succinctly, what is an accurate and efficient similarity metric for tactile geometry? With the ubiquity of LIDAR/RGB-D data, matching unordered point sets is vital for place recognition in the SLAM community [UL18; Kom21]. Following the seminal PointNet [Qi+17], Choy et al. [CGS19] later developed efficient and expressive sparse 3D convolution. This backbone was used to aggregate local point features to global point-cloud embeddings for LIDAR place-recognition [Kom21]. We show tactile geometries can be compacted into codes with the same architecture, for easy frame-to-model queries.

4.2 Problem formulation

For a vision-based tactile sensor dynamically sliding along a known object's surface, our goal is to track the distribution of 6D sensor pose $\mathrm{x}_t \in SE(3)$ in the object-centric frame. Such pose context is useful for downstream planning and control, for instance manipulating an object in-hand [Ope+18]. The sensor is affixed on a robot finger and we have access to the end-effector pose, but its relative position and orientation on the object's surface is unknown. At a each timestep t, our measurements are a tactile image, and noisy sensor pose in the robot's reference-frame, $\mathrm{z}_t = \left\{\mathrm{I}_t \in \mathbb{R}^{240\times320\times3},\ \mathrm{p}_t \in SE(3)\right\}$. For simplicity, we assume the object is stationary and sliding is a single continuous contact interaction. The dimensions of the objects are much larger than the sensor's contact area. We assume an uninformative initial prior for x_t, but later show the benefit of visual priors.

4.3 Global localization during sliding touch

MidasTouch comprises three distinct modules, as illustrated in Fig. 4.1: a tactile depth network (TDN: Section 4.3.1), tactile code network (TCN: Section 4.3.2), and particle filter (Section 4.3.3). At a high-level, the TDN first converts a tactile image I_t into its local 3D geometry P_t via a learned observation model. The 3D information is then condensed into a tactile code E_t by the TCN through a sparse 3D convolution network. Finally, the downstream particle filter uses these learned codes in its measurement model, and outputs a sensor pose distribution that evolves over time. Throughout this work, we use 10 YCB objects with diverse geometries in our tests: `sugar_box`, `tomato_soup_can`, `mustard_bottle`, `bleach_cleanser`, `mug`, `power_drill`, `scissors`, `adjustable_wrench`, `hammer`, and `baseball`.

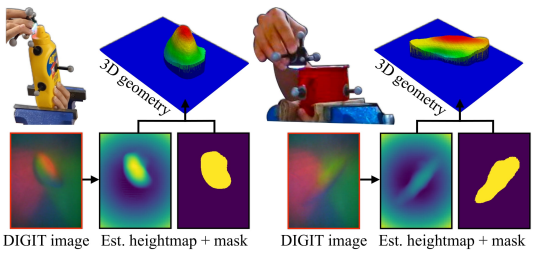


Figure 4.2: Real-world DIGIT images in the *YCB-Slide* dataset. Our tactile depth network (Section 4.3.1) is trained in simulation, and predicts 3D geometry given an input tactile image.

4.3.1 Tactile depth network (TDN)

The tactile depth network learns the inverse sensor model to recover local 3D geometry from a tactile image. We adapt a fully-convolutional residual network [Lai+16; Sur+22b], and supervise it to predict local heightmaps from tactile images.

Network and training: This network uses a ResNet-50 backbone, with a series of up-projection blocks. With the optical tactile simulator TACTO [Wan+22], we render a large collection of DIGIT images with ground-truth heightmaps. The images are from simulated interaction with 40 YCB [Cal+17] object meshes, while holding out all test objects. The images are rendered at 5000 different poses per object: randomizing for contact point, orientation, and indentation depth. For sim2real transfer, we calibrate TACTO with real-world data from multiple independent DIGITs, and apply random lighting augmentations. Per object, the split of train-validation-test is $4000:500:500$.

Image to 3D: At each timestep, we pass the 240×320 RGB tactile image I_t through the network to get heightmap H_t. To remove non-contact areas we compute a mask C_t by depth thresholding, and get $\hat{\mathrm{H}}_t = \mathrm{H}_t \odot \mathrm{C}_t$. Finally, the heightmap is reprojected to 3D via the camera's known perspective projection model $\hat{\mathrm{H}}_t \mapsto \mathrm{P}_t$. Evaluations on the test set show heightmap RMSE of 0.135 mm with respect to ground-truth. Fig. 4.2 shows examples of the TDN on real-world DIGIT interactions.

4.3.2 Tactile code network (TCN)

This network summarizes large, unordered point clouds of local geometry into a low-dimensional embedding space, or code. If two sensor measurements are nearby in pose-space, they observe similar geometries and therefore, their codes will also be nearby in embedding-space. However, the inverse is not necessarily true: measurements from two opposite corners of a cube may have similar codes, but their corresponding sensor poses are dissimilar. This is an inherent challenge of tactile localization, the so-called *contact non-uniqueness* highlighted by Bauza et al. [BBR22]. In our work getting the most-likely modes of this distribution is sufficient since the downstream particle filter can then disambiguate them temporally.

An analog in the SLAM community is the loop closure problem from 3D data. LIDAR place recognition modules, such as PointNetVLAD [UL18] and MinkLoc3D [Kom21], use metric learning to learn point cloud similarity. While tactile images differ from natural images, the geometries from LIDAR and tactile sensors (normalized for scale) are similar. The MinkLoc3D architecture, based on MinkowskiNet [CGS19], performs state-of-the-art point cloud retrieval through sparse 3D convolutions.

Network and training: The architecture comprises of three components: (i) Voxelization of P_t into a quantized sparse tensor $\hat{\mathrm{P}}_t = \{\langle \hat{x}_i, \hat{y}_i, \hat{z}_i, 0/1 \rangle\}$, (ii) Feature pyramid network [Lin+17]

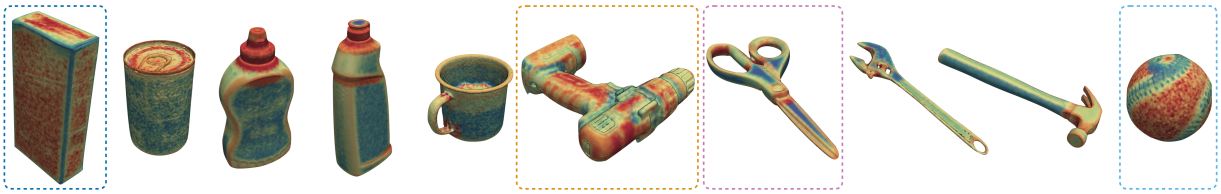


Figure 4.3: Tactile codebook per object visualized as a spectral colorspace map using t-SNE [VH08]. Each codebook comprises of 50k densely sampled poses with their corresponding 256-dimensional tactile code. Similar hues denote sensor poses that elicit similar tactile codes. We can clearly delineate local geometric features: edges (`sugar_box`) , ridges (`power_drill`), corners (`scissors`), and complex texture (`baseball`).

for per-voxel local features $\hat{\mathrm{P}}_t^f = \{\langle \hat{x}_i, \hat{y}_i, \hat{z}_i, \hat{\mathrm{f}}_i \in \mathbb{R}^{256} \rangle\}$, and (iii) Generalized-mean pooling for point cloud embedding vector $\mathrm{E}_t \in \mathbb{R}^{256}$. We use the pre-trained weights, learned from four comprehensive LIDAR datasets [Kom21], and fine-tune them with TACTO data collected over the 40 YCB objects. This is trained with a contrastive triplet loss, and we accumulate positive and negative local 3D geometries through pose-supervision.

Tactile codebook: Once we have a code E_t, we would like to efficiently compare this with a dense set of contacts. Inspired by prior work [Den+21; BBR22], we build a tactile codebook comprising of $M = 50$k randomized sensor poses on each object's mesh. We evenly sample these points and normals on the mesh with random orientations and indentations. We feed these poses into TACTO to generate a dense set of tactile images for each object.

We pass the generated images through the TDN + TCN to get a codebook for the specific object o: $\mathcal{C}_o = \{\langle \mathrm{p}^{[m]}, \mathrm{E}^{[m]} \rangle\}_{m=1}^{M}$, where $\mathrm{p}^{[m]}$ are $SE(3)$ sensor poses in codebook and $\mathrm{E}^{[m]}$ are corresponding tactile codes. Fig. 4.3 shows a t-SNE visualization of the codebooks, a colorspace representation of local geometric similarity. We observe regions with similar geometries have identical hues and the traversal of our sensor between these geometries will give us valuable measurement signals.

We build a KD-Tree with 6-element vectors: $\{[\mathrm{p}_{\text{trans}}^{[m]}, \ \alpha \log(\mathrm{p}_{\text{rot}}^{[m]})]\}_{m=1}^{M}$ for nearest-neighbor search. Here, $\log(\cdot)$ is the $SO(3)$ logarithm map obtained via Theseus [Pin+22], and $\alpha = 0.01$ is the rotation scaling factor. Thus, given a candidate pose $\mathrm{x}_t^{[i]}$ we do not have to render and obtain its corresponding code, but instead just perform a pose-space lookup $\mathcal{C}_o(\mathrm{x}_t^{[i]})$. With memoization of code generation, we make getting codes for thousands of arbitrary pose particles tractable.

Single-touch localization: To understand the effectiveness of codes as a proxy for pose prediction, we conduct a set of single-touch experiments in simulation. This also provides an idea of which objects are salient, and which are adversarial, as highlighted in Fig. 4.4. We observe that we perform significantly better than random touches for all objects, and those with symmetric, regular structures exhibit long-tail errors. This shows that while a single-touch has meaningful signal, it can be valuable to disambiguate these readings temporally with a filtering framework.

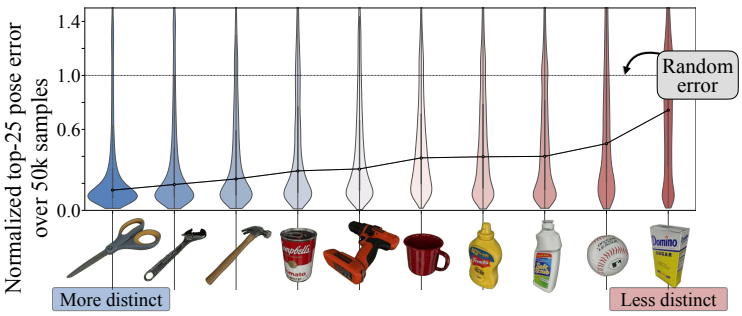


Figure 4.4: Pose-error for 50k single-touch queries on YCB objects, in ascending order (normalized with respect to random touch). For each query, we get the top-25 highest scores from the tactile codebook $\mathcal{C}_o$, and compute their minimum pose-error with respect to ground-truth. We observe tools with salient geometries to be easier to localize versus objects that exhibit symmetry.

4.3.3 Filtering over posed touch

The particle filter approximates the posterior distribution of the sensor pose as a set $\mathrm{X}_t = \{\langle \mathrm{x}_t^{[i]}, w_t^{[i]} \rangle\}_{i=1}^{N_t}$, where $\mathrm{x}_t^{[i]}$ are $SE(3)$ sensor pose particles, $w_t^{[i]}$ are the predicted particle weights, and N_t is number of particles. Rather than a unimodal Gaussian, this arbitrary distribution captures the multi-modality of global localization. We propagate the posterior distribution based on the stream of tactile images and noisy sensor poses, emulating a conventional particle filter but with a learned measurement model.

Initialization: The initial distribution X_0 is sampled from a coarse prior, which can (optionally) be obtained from vision [Den+21]. Estimating contact location from vision is inherently noisy and can be ambiguous due to object symmetries. We find that even with very uninformative priors, our multi-modal filter can prune out hypotheses and converge to the correct one.

We sample particles from a 6D prior about the ground-truth pose x_0^{gt} as $\mathrm{x}_0 \sim \mathcal{N}(\mathrm{x}_0^{\text{gt}}, \begin{bmatrix} \sigma_{\text{trans}} & 0 \\ 0 & \sigma_{\text{rot}} \end{bmatrix})$, such that $3\sigma_{\text{trans}} = \mathcal{M}_{\text{diag}}$ and $3\sigma_{\text{rot}} = 180°$. This weak prior scatters the particles across the object surface, and allows comparison across objects as a function of their mesh diagonal length $\mathcal{M}_{\text{diag}}$. In our results, we also show particle filter ablations with tighter pose priors. After sampling, we project the particles back onto the surface by querying their nearest neighbors from the tactile codebook.

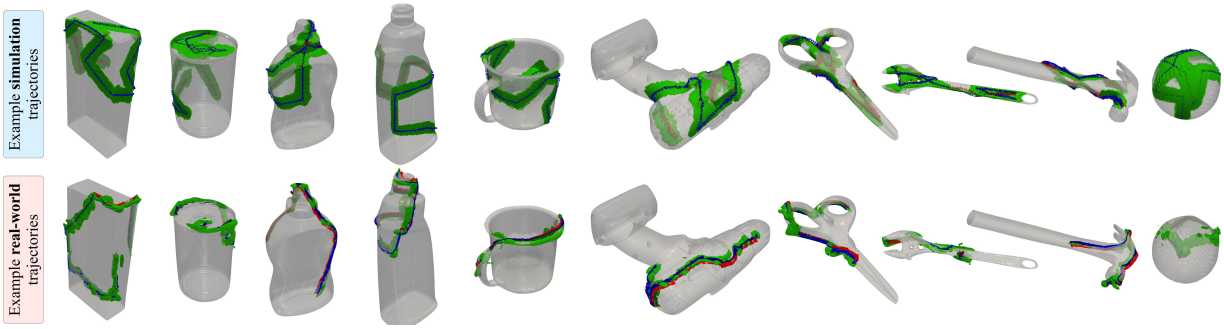


Figure 4.5: Example sliding trajectories from simulated and real trials on the 10 YCB objects. Overlaid in green are the local 3D geometries captured by the tactile sensor, and the contact poses as RGB coordinate axes.

Motion model: We typically have access to noisy global estimates of end-effector pose, p_t, via robot kinematics. Our sensor odometry comes from the relative pose predictions $\Delta p_t = (p_{t-1})^{-1} \cdot p_t$. The motion model propagates particles X_{t-1} forward by sampling from a state transition probability:

$$\mathrm{x}_t^{[i]} \sim p(\mathrm{x}_t \mid \mathrm{x}_{t-1}^{[i]}, \Delta p_t) = \mathrm{x}_{t-1}^{[i]} \oplus \mathcal{N}(\Delta p_t, \Sigma_{\Delta p}) \tag{4.1}$$

Odometry could also be from marker flow [Yua+15] or frame-to-frame tracking [Sod+21; Sod+22b].

Measurement update: We update the particle weights based on how their tactile codes match the current measurement E_t, as a function of cosine distance. At each timestep, we lookup the codebook for particle codes (Section 4.3.2) and perform a matrix-vector multiplication:

$$w_t^{[i]} = p(\mathrm{z}_t \mid \mathrm{x}_t^{[i]}) \sim \text{softmax}\left(\frac{\mathrm{E}_t \cdot \mathcal{C}_o(\mathrm{x}_t^{[i]})}{||\mathrm{E}_t|| \cdot ||\mathcal{C}_o(\mathrm{x}_t^{[i]})||}\right) \tag{4.2}$$

These weights are scaled $[0, 1]$ and represent how well the candidate particle poses match with the current measurement. These weights feed into the subsequent resampling step.

Particle resampling: We sample a new set of particles X_t from the proposal distribution with a probability proportional to their weights. Low-variance resampling [Thr02b] is a popular solution that covers the sample set in a systematic manner.

Hypothesis clustering: Alongside a distribution of poses, downstream tasks may need distinct pose hypotheses. To achieve this, we hierarchically cluster particles in $\mathbb{R}^3$ using DBSCAN [Sch+17a]. We average the cluster positions and quaternions [Mar+07] to get a hypothesis set $h_t : \{\mathrm{x}_t^1 \ldots \mathrm{x}_t^H\}$.

4.4 The *YCB-Slide* dataset

To evaluate MidasTouch, and enable further research in tactile sensing, we introduce our *YCB-Slide* dataset. It comprises of DIGIT sliding interactions on the 10 YCB objects from Section 4.3. We envision this can contribute towards efforts in tactile localization, mapping, object understanding, and learning dynamics models. We provide access to DIGIT images, sensor poses, ground-truth mesh models, and ground-truth heightmaps + contact masks (simulation only).

Figure 4.6: Real-world sliding trials in the *YCB-Slide* dataset. Inset is an example tactile image from the interactions, capturing the local geometry of the object.

Simulated interactions: We simulate sliding across objects using TACTO, mimicking realistic interaction sequences. The pose sequences are geodesic-paths of fixed length $L = 0.5$m, connecting random waypoints on the mesh. We corrupt sensor poses with zero-mean Gaussian noise

$\sigma_{\text{trans}} = 0.5\text{mm}$, $\sigma_{\text{rot}} = 1^\circ$. We record five trajectories per-object; 50 interactions in total. A representative sample of the trajectories, along with their accrued tactile geometries are shown in Fig. 4.5.

Real-world interactions: In the real-world, we perform sliding experiments through handheld operation of the DIGIT. We keep each YCB object stationary with a heavy-duty bench vise, and slide along the surface and record 60s trajectories at 30Hz. We use an OptiTrack system for timesynced sensor poses, with 8 cameras tracking the reflective markers. We affix six markers on the DIGIT and four on each test object. The canonical object pose is adjusted to agree with the ground-truth mesh models. Minor misalignment of sensor poses from human error are rectified by postprocessing the trajectories to lie on the object surface. We record five logs per-object, for a total of 50 sliding interactions. Our experimental setup and collected data are illustrated in Figures Fig. 4.5 and Fig. 4.6.

4.5 Experimental results

Simulation: We evaluate MidasTouch over the 50 trajectories collected in Section 4.4. As the filter is non-deterministic, each trajectory is run 10 times for averaged results over 500 trials.

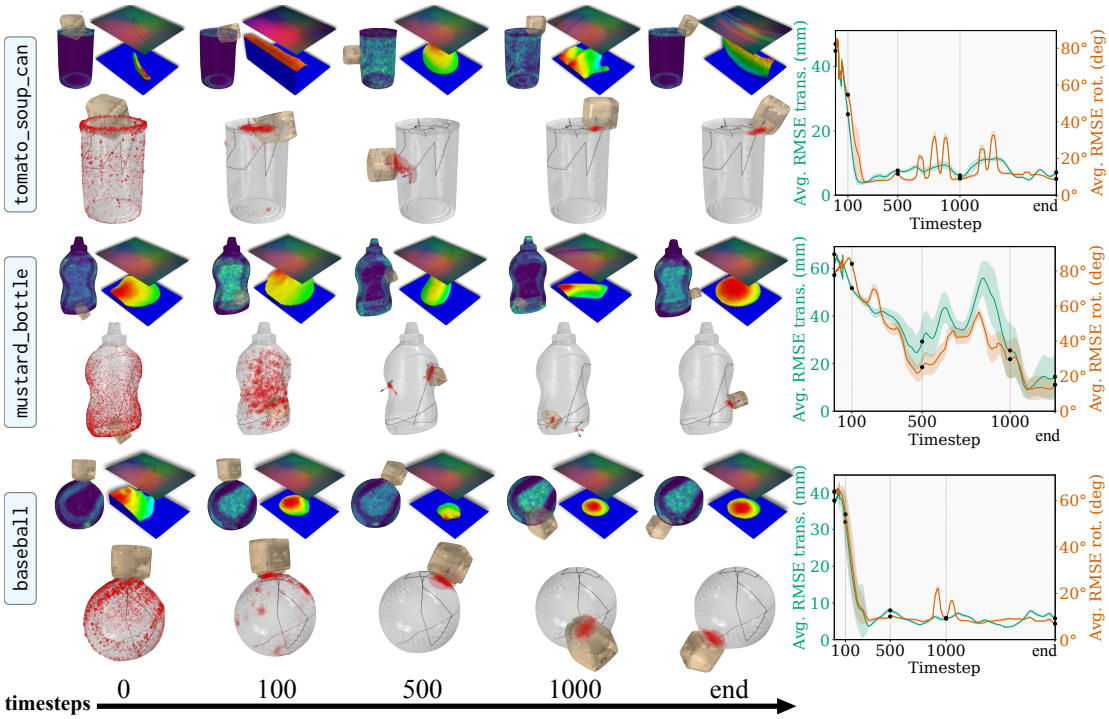


Figure 4.7: Snapshots of simulated sliding results on three YCB objects. For each row: **[top]** the tactile images, local geometries, and heatmap of pose likelihood with respect to the tactile codebook, **[bottom]** pose distribution evolving over time, and converging to the most-likely hypothesis after encountering salient geometries, **[right]** average translation and rotation RMSE of the distribution over time with variance over 10 trials.

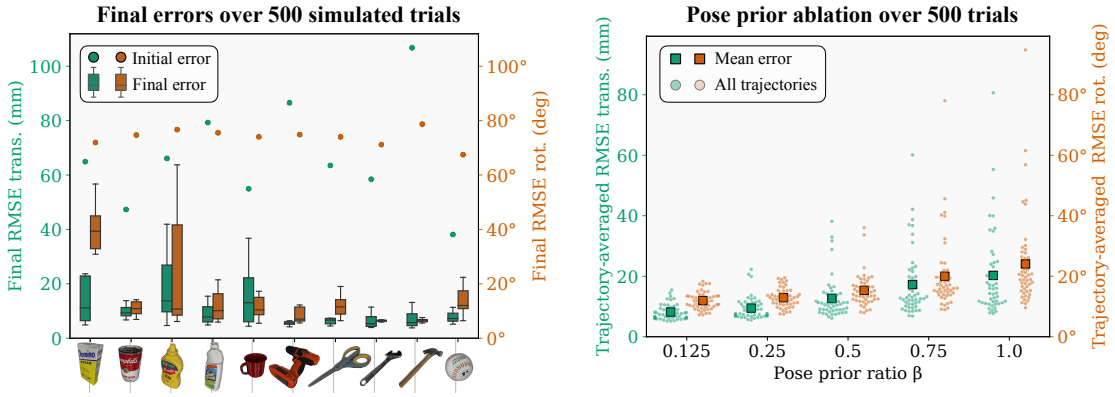


Figure 4.8: [left] Boxplot of final trajectory error over 500 simulation trials. For each object, we plot the averaged initial and final RMSE for the particle set. We observe better convergence for tools with salient geometries, as opposed to symmetric objects. **[right]** Ablation over initial pose uncertainty, to show lower average trajectory error with better visual priors. With a weaker initialization ($\beta = 1.0$), outliers in pose-error are more prevalent.

Fig. 4.7 shows qualitative results for three representative trajectories. At each timestep, we visualize the pose distribution and plot the average particle RMSE with respect to the ground-truth pose. We see convergence to the most-likely pose hypothesis over the sliding interactions. Upon convergence, the hypothesis clustering step sets an averaged pose for each cluster: visualized

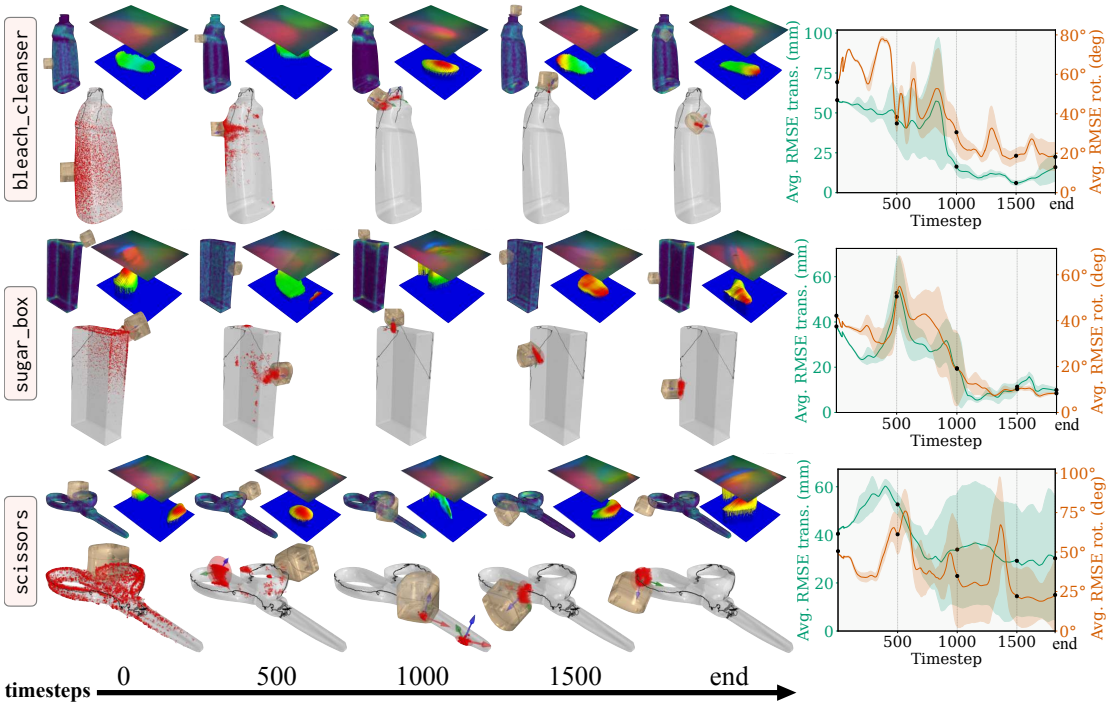


Figure 4.9: Snapshots of real-world sliding results on three YCB objects. For each row: **[top]** the tactile images, local geometries, and heatmap of pose likelihood with respect to the tactile codebook, **[bottom]** pose distribution evolving over time, and converging to the most-likely hypothesis after encountering salient geometries, **[right]** average translation and rotation RMSE of the distribution over time with variance over 10 trials.

as the pose axes with uncertainty ellipses. We also highlight the current tactile image, surface geometry, and comparison with the tactile codebook. These trials are initialized with pose prior $[\sigma_{\text{trans}}, \sigma_{\text{rot}}]$ from Section 4.3.3.

Fig. 4.8 [left] accumulates quantitative metrics over the 500 simulated trials. First, we show the final pose errors across all trials compared against the initial error. Overall, the averaged final pose errors are 0.74cm and 9.43°; the per-object errors roughly correlate to the single-touch errors from Fig. 4.4. While the error drops significantly for most trials, it fluctuates depending on each trajectory's salient geometries (or lack thereof). Larger pose-errors can be attributed to tracking multiple modes that equally explain the same sliding sequence (please refer to supplementary video). We consider this a benefit of our multi-modal framework, and is especially prevalent for symmetric objects like the `sugar_box`, `mustard_bottle`, and `mug`.

We further perform ablations over pose prior (Fig. 4.8 [right]), initializing each trial with uncertainty $\beta \times [\sigma_{\text{trans}}, \sigma_{\text{rot}}]$. This serves as a proxy for visual-priors; better initializations lead to fewer candidate modes.

Real-world: We run 500 similar trials for the real-world data from Section 4.4 with $\beta = 0.5$, which serves as a bound for a reasonable prior estimate available from vision. The real-world tactile heightmaps are noisier than simulation, thus a tighter initialization prevents particle depletion. In Fig. 4.9 we present three representative trajectories that show the hypotheses converge to the ground-truth pose. The accumulated statistics in Fig. 4.10 shows reduced final error across all objects, except `baseball`. The averaged final pose errors are higher than the simulation results: 1.97cm and 21.48°. Once again, we see tools and intricate objects are the easiest to localize on while symmetric objects are the hardest. An anomaly is the `baseball`, on which we fail to localize in all trials. We attribute this to a lack of distinct geometry in real tactile images, effectively meaning we are trying to localize on a featureless sphere. These failure modes are highlighted in Fig. 4.11.

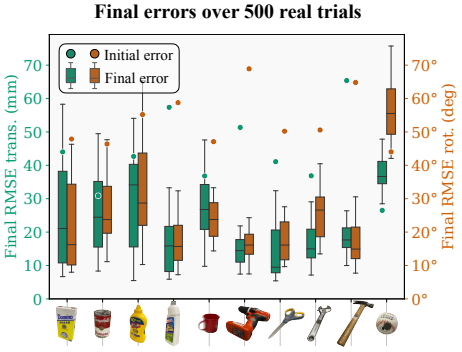


Figure 4.10: Boxplot of final error over 500 real-world trials from the 50 *YCB-Slide* trajectories.

4.6 Discussion and limitations

In this section, we demonstrate finger-object global localization from posed tactile images. This online method outputs a pose-distribution on the object's surface that converges over time as the sensor traverses salient surface geometries. Specifically, our system is the first to learn tactile embeddings from local 3D geometry, and disambiguate them with a nonparametric particle filter. Our experiments demonstrate the surprising effectiveness of a pairing learned tactile perception

modules with Monte-Carlo methods to resolve distribution ambiguities.

Currently, we are limited to a moving sensor relative to a fixed-pose object (or vice-versa). In future work, we wish to incorporate a dynamic object in our motion model through **(i)** visual measurements [Den+21], and/or **(ii)** local in-hand tracking [Sod+22b]. Our method currently cannot work when we lack ground-truth object models. Along with reconstructing objects [Sur+21; Sur+22b], we would need to build the tactile codebook on-the-fly. For future in-hand manipulation tasks, it is also necessary to scale MidasTouch to multi-contact configurations. Our on-surface assumption leads to poor behavior when we break contact with the object, and the effects of shear on the tactile image sequence have been ignored [ABL19]. Object compositionality doesn't hold for deformable or articulated objects, so modeling these properties is an interesting future direction [Wi+22a]. With a differentiable filter [JRB18] we can fine-tune end-to-end for more robust real-world performance.

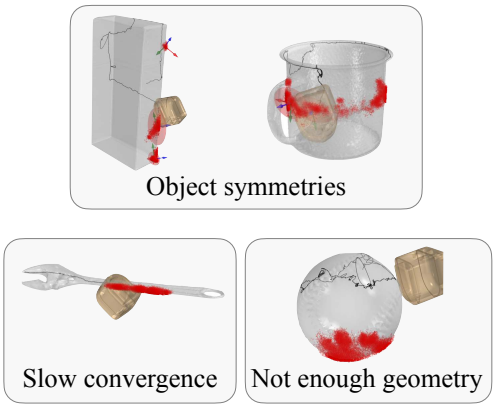


Figure 4.11: Failure modes on real-world trials: **(i)** it may be hard to converge to the true hypothesis for objects with symmetries, **(ii)** slow convergence of the filter can lead to large pose uncertainty, **(iii)** lack of discernible geometry can result in drift from the true mode.

Chapter 5

Multimodal in-hand perception

NeuralFeels

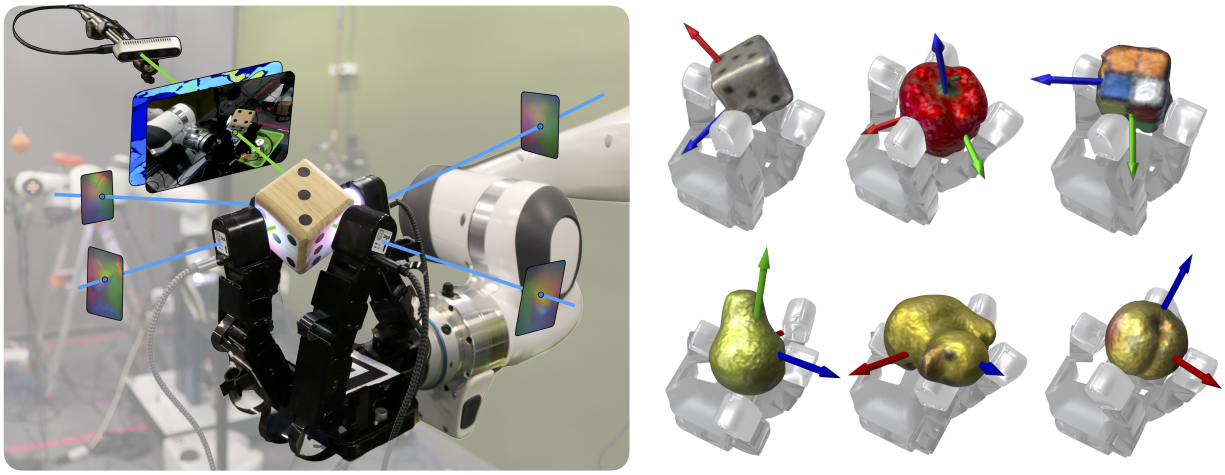


Figure 5.1: Visuo-tactile perception with NeuralFeels. Our method estimates pose and shape of novel objects **(right)** during in-hand manipulation, by learning neural field models online from a stream of vision, touch, and proprioception **(left)**.

To perceive deeply is to have sensed fully. Humans effortlessly combine their senses for everyday interactions—we can rummage through our pockets in search of our keys, and deftly insert them to unlock our front door. Currently, robots lack the cognition to replicate even a fraction of the mundane tasks we perform, a trend summarized by Moravec's Paradox [Mor88]. For dexterity in unstructured environments, a robot must first understand its spatial relationship with respect

This chapter is adapted from the pre-print *Neural feels with neural fields: Visuo-tactile perception for in-hand manipulation* by Sudharshan Suresh, Haozhi Qi, Tingfan Wu, Taosha Fan, Luis Pineda, Mike Lambeta, Jitendra Malik, Mrinal Kalakrishnan, Roberto Calandra, Michael Kaess, Joe Ortiz, and Mustafa Mukadam. *Under review*, arXiv preprint arXiv:2312.1346, Dec. 2023

to the manipuland. Indeed, as robots move out of instrumented labs and factories to cohabit our spaces, there is a need for generalizable spatial AI [Dav18].

Specific to in-hand dexterity, *knowledge of object pose and geometry* is crucial to policy generalization [Ope+18; Ope+19; Han+22; Qi+23]. As opposed to end-to-end supervision [Yin+23; Guz+23b; Che+23a], these methods require a persistent 3D representation of the object. However, the status quo for in-hand perception is currently restricted to the narrow scope of tracking known objects with vision as the dominant modality [Han+22]. Further, it is common for practitioners to sidestep the perception problem entirely, retrofitting objects and environments with fiducials [Ope+18; Ope+19]. To further progress towards general dexterity, it is clear that one of the missing pieces is general, robust perception.

With visual sensing, researchers tend to tolerate interaction rather than embrace it. This is at odds with contact-rich problems where self-occlusions is imminent, like rotating [Qi+22], reorienting [Han+22; Che+23a], and sliding [She+21; Sur+22a]. Additionally, vision often fails in the real-world due to poor illumination, limited range, transparency, and specularity. Touch provides a direct window into these dynamic interactions, and human cognitive studies have reinforced the complementarity with vision [HE07].

Hardware advances have led to affordable vision-based touch sensors [YDA17; Don+18; War+18; Als+19; Lam+20; Pad+20; Wan+21] like the GelSight and DIGIT. Progress in touch simulation [Wan+22] enables practitioners to learn tactile observation models that transfer to real-world interactions [Wan+21; Sod+22b; Sur+22b]. With a fingertip form-factor, their illuminated gel deforms on contact and the physical interaction is captured by an internal camera. When chained with robot kinematics, we obtain dense, situated contact that can be processed similar to natural camera images.

Now given multimodal sensing, how best to represent the spatial information? Coordinate-based learning, formalized as *neural fields* [Xie+22], has found great success in visual computing. With neural fields, practitioners can create high-quality 3D assets offline given noisy visual data and pose annotation [Mil+21; Mül+22; Li+23]. They are continuous representations with higher fidelity than their discrete counterparts like point clouds and meshes. While they are specialized towards batch optimization, lightweight SDF models [Ort+22; Suc+21; Zhu+22; Wen+23] have made online perception possible.

Researchers have used this extensible architecture not just for continuous 3D quantities like signed distance fields (SDFs) and radiance [Par+19; Mil+21; Mül+22], but also for pose estimation [Yen+21; Wen+23], planning [Gro+23], and latent physics [Le +23]. Moreover, the ease of imparting generative priors [Yu+21] and initializing with pre-trained models [Par+19] future-proofs them. While neural fields have emerged little by little in robot manipulation [Zho+22; Ker+22a; Wi+22b; Gro+23], the optimization of multimodal data remains an open question.

NeuralFeels presents an online solution to localize and reconstruct objects for in-hand manip-

ulation with multimodal sensing. We unify vision, touch, and proprioception into a neural representation and demonstrate SLAM for apriori unknown objects, and robust tracking of known objects. In our experiments, we present our robot with a novel object, and it infers and tracks its geometry through just interaction. We use a dexterous hand [Won23] sensorized with commercial vision-based touch sensors [Lam+20] and a fixed RGB-D camera (Figure 1.1). With a proprioception-driven policy [Qi+22] we explore the object's extents through in-hand rotation.

Through our experiments we study the role that vision and touch play in interactive perception, the effects of occlusion, and visual sensing noise. To evaluate our work, we collect a dataset of 70 in-hand rotation trials in both the real-world and simulation, with ground-truth object meshes and tracking. Our results on novel objects show average reconstruction F-scores of 81% with pose drifts of just 4.7 mm, further reduced to 2.3 mm with known CAD models. Under heavy occlusion, we demonstrate up to 94% improvements in pose tracking compared to vision-only methods. Our combination of rich sensing and spatial AI requires minimal hardware compared to complex sensing cages, and is easier to interpret than end-to-end perception methods. The output of the neural SLAM pipeline—pose and geometry—can drive further research in general dexterity, broadening the capabilities of home robots.

The proposed research puts together the puzzle pieces from prior chapters: both Section 3 and Section 4 assumed fixed object pose in the 3D SLAM task. We relax this assumption in Section 2, but for a 2D problem, where we reconstruct a moving planar object. With these insights, we wish to tackle a multi-finger SLAM task via in-hand manipulation.

5.1 Related work

The domain of our work—an intersection of simultaneous localization and mapping (SLAM) and manipulation—has been studied for over two decades. A first exemplar is from Moll and Erdmann [ME04], who reconstruct the shape and motion of an object rolled between robot palms, later reproduced with specialized sensors [Str+14; Lep+23]. Tactile SLAM has been thoroughly investigated for planar pushing due to its well-understood mechanics [YLR15; Sur+21]. The

iMAP **[Suc+21]** NICE-SLAM **[Zhu+22]** iSDF **[Ort+22]**

Figure 5.2: Online learning has been applied in RGB-D SLAM for indoor and tabletop scenes. iSDF [Ort+22] has been shown to reconstruct room-scale and tabletop scenes from posed depth images. iMAP [Suc+21] and NICE-SLAM [Zhu+22] also perform camera tracking via the differentiable renderer, in conjunction with SDF optimization.

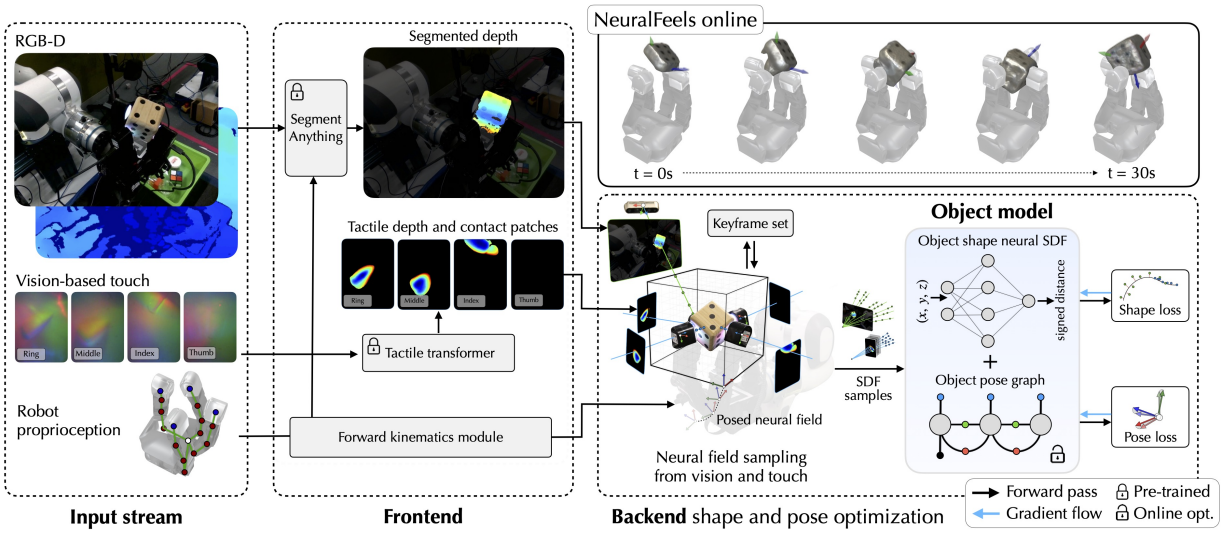


Figure 5.3: A visuo-tactile perception stack amidst interaction. An online representation of object shape and pose is built from vision, touch, and proprioception during in-hand manipulation. Raw sensor data is first fed into the *frontend*, which extracts visuo-tactile depth with our pre-trained models. Following this, the *backend* samples from the depth to train a neural signed distance field (SDF), while the pose graph tracks the posed neural field.

combination of vision and touch has been explored for reconstructing fixed objects [Wan+18; Smi+20; Sur+22b; Che+23b] and tracking known objects [YR18; Lam+19; Sod+21]. Closest to our work is FingerSLAM [ZBA23], combining dense touch from a single finger with vision, however we consider the more challenging case of in-hand manipulation.

5.2 Problem formulation

NeuralFeels ingests multimodal information to build a persistent 3D object representation. Similar to classical SLAM frameworks, it first has a *frontend*, responsible for abstracting the vision (RGB-D) and touch (RGB) input stream into a format suitable for estimation (segmented depth). Thereafter, the *backend* fuses this data into an optimization structure that infers the *object model*: an evolving posed object SDF. An illustration of the entire pipeline is found in Figure 4.1, which we refer the reader back to throughout this section.

NeuralFeels incrementally builds an *object model*, simultaneously optimizing for the object SDF network's weights θ and its corresponding pose x_t at the current timestep t. For object exploration, we use a proprioception-driven policy π_t that executes the optimal action to achieve stable rotation. The input stream of all sensors $\mathcal{S}$ consists of the following (left column of Figure 4.1):

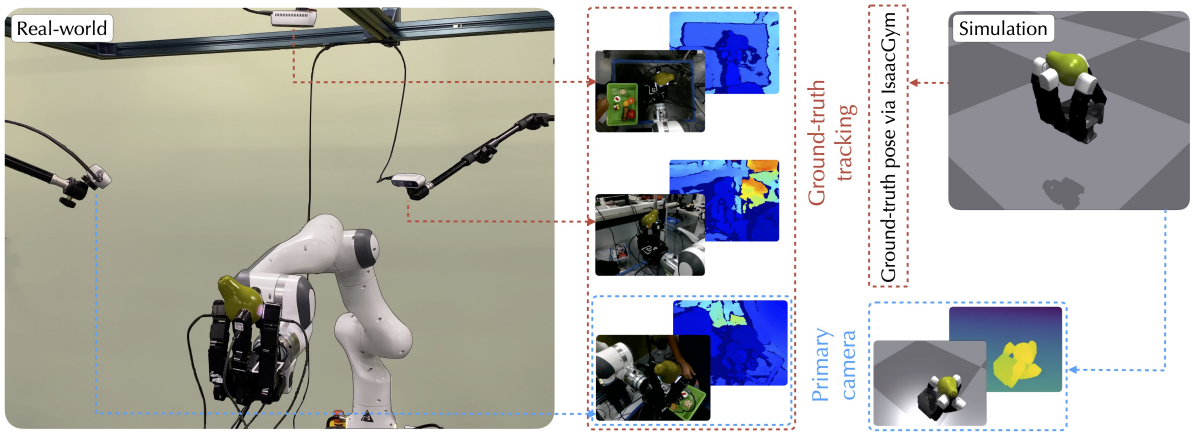
- **RGB-D vision**: image I_t^c and depth D_t^c from calibrated camera $c \in \mathcal{S}$
- **RGB touch**: images I_t^s from four DIGITs [Lam+20]; $s \in \{d_{\text{index}}, d_{\text{middle}}, d_{\text{ring}}, d_{\text{thumb}}\} \in \mathcal{S}$
- **Proprioception**: joint-angles q_t from robot encoders.

5.3 Robot setup

5.3.1 Hardware and simulation

The Allegro hand [Won23] is retrofit with four DIGIT vision-based tactile sensors [Lam+20], at each of the distal ends. The DIGIT produces a 240×320 RGB image of the physical interaction at 30 Hz. The Allegro publishes 16D joint-angles so as to situate the tactile sensors with respect to the base frame. The hand is rigidly mounted on a Franka Panda arm, with an Intel D435 RGB-D camera placed at approximately 35 cm from it. The camera extrinsics are computed with respect to the base frame of the Allegro through ArUco [Gar+14] hand-eye calibration. For our vision pseudo-ground-truth we use three such cameras in the workspace (Figure 5.4), jointly calibrated via Kalibr [FRS13], to achieve ≈ 1 px reprojection error. Our simulator replicates the real-world setup: a combination of the IsaacGym physics simulator [Mak+21] with the TACTO touch renderer [Wan+22]. In this case, we can record and store the true ground-truth object pose directly from IsaacGym.

(a) FeelSight dataset: real-world and simulated object interactions



(b) Robot setup in the real-world and simulation

Figure 5.4: Robot setup in the real-world and simulation. (a) We capture diverse visuo-tactile interactions across different object categories in the real-world and physics simulation. **(b)** The robot cell is made up of three realsense RGB-D cameras, an Allegro robot hand mounted on a Franka Panda, and four DIGIT tactile sensors. All real-world results use the primary camera and DIGIT sensing, while the additional cameras are fused for our ground-truth pose tracking. In simulation, we use an identical primary camera in IsaacGym with touch simulated in TACTO. The simulator provides ground-truth object pose, so multi-camera tracking is not necessary.

5.3.2 FeelSight: a visuo-tactile perception dataset

Visuo-tactile perception lacks a standardized benchmark or dataset that has driven progress in adjacent fields like visual tracking [Hod+18], SLAM [Gei+13], and reinforcement learning [Jam+20]. Towards this, we introduce our FeelSight dataset for visuo-tactile manipulation. We use the in-hand rotation policy (Section 5.3.3) to collect vision, touch, and proprioception for 30 seconds per trial.

When we encounter a novel object, we tend to twirl it in our hand to get a better look from different views, and regrasp it from different angles. The equivalent for a multi-fingered hand, in-hand rotation, is an ideal choice for the interactive perception problem. We adopt the method of Qi et al. [Qi+22] where they train a proprioception-based policy in simulation, and directly transfer it to the real-world. Recent work has further shown in-hand object rotation using touch and proprioceptive history [Qi+23; Yin+23], however our simpler abstraction proves sufficient for this task. In our experiments, the rotation policy π_t sends commands to the robot hand at 20 Hz via the ROS Allegro controller. This achieves stable rotation of novel objects and interesting visuo-tactile stimuli; for further details refer to Section 5.3.3.

The dataset has 5 in-hand rotation trials each of 6 objects in the real-world and 8 objects in simulation; a total 35 minutes of interaction. As explained in Figure 5.4, we record a pseudo-ground-truth in the real-world, and exact ground-truth poses in simulation. We ensure diversity in the class of objects: they vary in geometry and size from 6-18 cm in diagonal length. Ground-truth meshes of each object are obtained with the Revopoint 3D scanner [Rev23], which uses dual-camera infrared for ≈ 0.05 mm scan accuracy. Additionally, the the simulated experiments have ground-truth meshes from the YCB [Cal+17] and ContactDB [Bra+19] datasets.

5.3.3 In-hand exploration policy

We first train a policy in simulation with access to an embedding of physical properties such as object position, size, mass, friction, and center-of-mass (denoted as z_t). From the joint-angles q_t and this embedding z_t, the policy outputs a PD controller target $a_t \in \mathbb{R}^{16}$. The policy is trained in parallel simulated environments [Mak+21] using proximal policy optimization [Sch+17b]. The reward function is a weighted combination of a rotational reward, joint-angle regularizer, torque penalty, and object velocity penalty. The resulting policy can adaptively rotate objects in-hand according to different physical properties.

During deployment, however, the policy does not have access to these physical properties. The estimator is instead trained to infer z_t from a history of proprioceptive states, which is in turn fed into the policy π_t. A crucial change compared to Qi et al. [Qi+22] is that we train the policy to rotate objects with DIGIT sensors on the distal ends (Figure 1.1). This results in different gaits, as it **(i)** relies on finger-object friction instead of gravity, and **(ii)** learns to maintain contact with the DIGIT gelpads.

5.4 Method overview and insights

Object model (Section 5.5): We represent the object SDF as a neural network with weights θ, whose output is transformed by the current object pose $\mathrm{x_t}$. This continuous function $F^{\theta}_{\mathrm{x_t}}(\mathrm{p}) : \mathbb{R}^3 \rightarrow \mathbb{R}$ maps a 3D coordinate p to a scalar signed-distance from the object's closest surface. Online updates are decomposed into alternating steps between refining the weights of the neural SDF θ, and optimizing the object pose $\mathrm{x_t}$. Our bespoke object model is a representation of both the pose and object geometry over time.

Frontend (Section 5.6): Given the RGB-D, RGB, and proprioception inputs, our frontend returns segmented depth measurements compatible with our backend optimizer. These modules are pre-trained with a large corpus of data.

Shape optimizer (Section 5.7.1): Takes in frontend output and optimizes for θ at fixed object pose $\mathrm{x_t}$ via gradient descent [Mül+22]. Each shape iteration results in improved object SDF $F^{\theta}_{\mathrm{x_t}}$.

Pose optimizer (Section 5.7.2): Builds and optimizes an object pose-graph [Pin+22] for x_t given fixed network weights $\bar{\theta}$. Every pose iteration spatially aligns the evolving object SDF with the current set of frontend output.

Insight 1: NeuralFeels is a posed neural field

The object model $F^{\theta}_{\mathrm{x_t}}$ is estimated by a chicken-and-egg optimization of both the instant-NGP weights θ, and the object pose x_t. Prior work has estimated the pose of a sensor in fixed neural field, either by freezing the network weights [Yen+21; Lin+21], or joint-optimization [Suc+21; Zhu+22; RLC22]. In our case, robot kinematics gives us the pose of the touch sensors, and extrinsics give us the pose of the camera. So, we instead *flip* this paradigm to estimate the pose of the neural field with respect to known-pose sensors.

Insight 2: Touch is vision, albeit local

We extend neural fields to directly incorporate touch just as it would vision. Our key insight is that vision-based touch can be approximated as a perspective camera model in tactile simulators like TACTO [Wan+22]. There are, however, differences that must be accounted for in image formation **(i)** vision-based tactile sensor impose their own color and illumination to the scene, which makes it hard to get reliable visual cues, **(ii)** a tactile image stream has considerably smaller metric field-of-view and depth-range is usually in centimeters rather than meters, **(iii)** tactile images have depth discontinuities along *all* non-contact regions, while natural images only encounter them along occlusion boundaries. Our method adapts each of these by **(i)** consistently using depth rather than color for optimization, **(ii)** sampling at different scales (centimeter v.s. meter) based on sensing source, **(iii)** sampling only surface points for touch, but both free-space and surface points for vision. More details are described in Section 5.7.1. After accounting for

these differences, we can sample touch consistent with vision, giving us a rich perspective of the object.

5.5 Object model

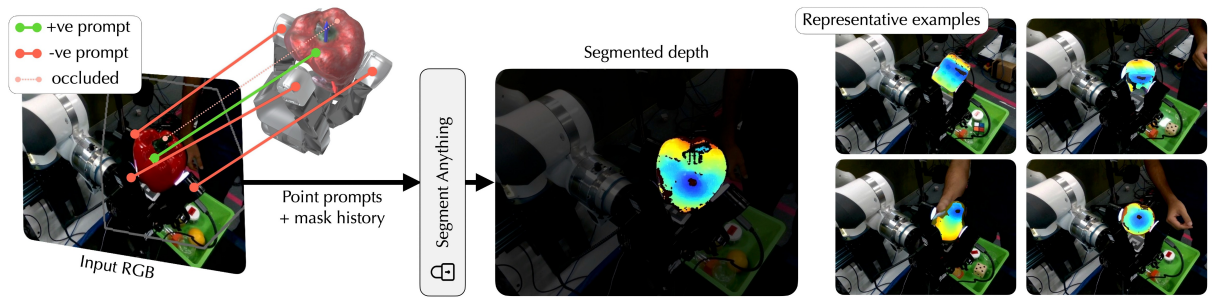
Our object model is depicted in the right column of Figure 4.1. In general, a neural SDF [Ort+22; Azi+22; Mül+22] represents 3D surfaces as the zero level-set of a learnable function $F(\mathrm{p}) : \mathbb{R}^3 \rightarrow \mathbb{R}$. The scalar field's sign indicates if any query point p in the volume is inside (negative), outside (positive) or on (≈ 0) the reconstructed surface. p is first *positionally-encoded* [Tan+20] into a higher-dimensional space, an important routine that helps networks better approximate high-frequency surfaces. This is followed by a multi-layer perceptron (MLP) that fits the encoding to a scalar field. Typically, this network is optimized with depth samples from a camera of known intrinsics, and annotated poses from structure-from-motion [SF16].

A neural SDF is more compact than the more popular neural radiance fields [Mil+21], as they do not model color and appearance properties of the scene. This is sufficient for manipulation, as we care more about estimating geometry than generating novel-views. Recently, instant-NGP [Mül+22] demonstrated a learnable multiresolution hash table as a positional encoding that greatly accelerates SDF optimization with small MLP backbones. This has been successfully leveraged for real-time SLAM in indoor scene [RLC22].

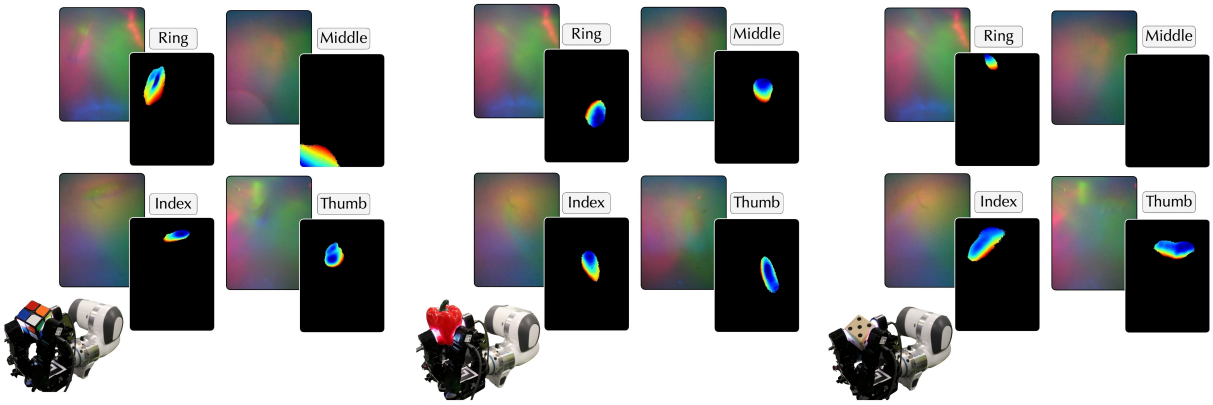
In our work, $F^{\theta}_{\mathrm{x}_t}$ represents the neural SDF of the object at a given pose x_t. While x_t is initialized to be between the robot fingers, θ is randomly initialized. Both shape and pose are estimated via alternating optimization, which emulating the paradigm of tracking and mapping that has found great success in robot vision [Cad+16]. The model is fully-differentiable, can be queried arbitrarily in 3D space, and easily extensible to color, latent physics, and other properties.

5.6 Frontend

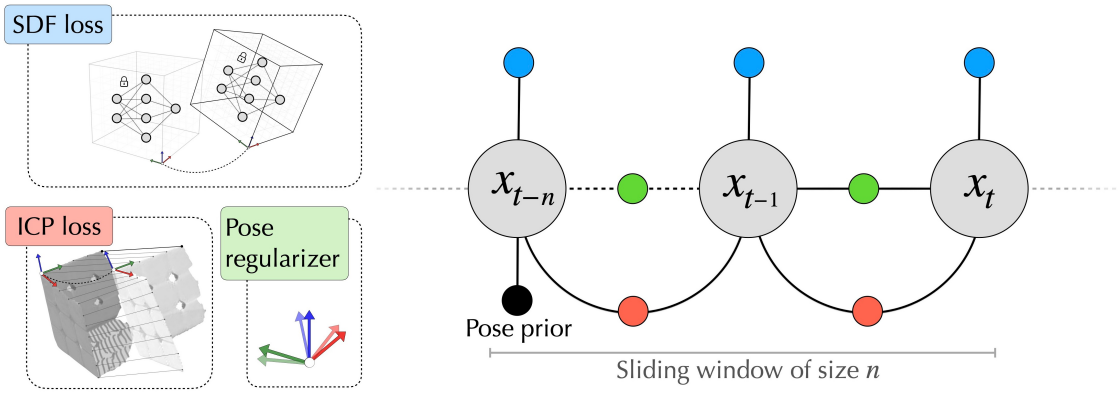
The frontend processes are shown in the center column of Figure 4.1. Its function is to robustly extract depth measurements from raw vision and touch sensing. Depth is available as-is in an RGB-D camera, but the challenge is to robustly segment out object depth pixels in heavily-occluded interactions. Towards this, we introduce a kinematics-aware segmentation strategy using powerful vision foundation models [Kir+23] (Section 5.6.1). Estimating depth from vision-based touch is an open research problem [BCR19; Wan+21; Sod+22b; Amb+21; Sur+22b] where millimeter precision and generalization across sensors is important. Towards this, we present a transformer architecture that accurately predicts DIGIT contact patches from inputs images (Section 5.6.2). Unlike our backend that is optimized online, the frontend networks are pre-trained from a large corpus of data. The output of our frontend is a segmented depth image $\hat{D}^s_t$ for each sensor $s \in \mathcal{S}$.



(a) Frontend: visual segmentation



(b) Frontend: tactile transformer



(c) Backend: pose graph optimization

Figure 5.5: Frontend and backend description. **(a)** Segment-Anything [Kir+23] combined with embodied prompts, gives us robust object segmentation. Through reasoning about finger occlusion and object pose with respect to the fingers, we can accurately prompt the segmentation network for robust output masks. **(b)** Representative examples of the sim-to-real performance of the tactile transformer. Each RGB image is fed through the network to output a predicted depth, along with a contact mask. **(c)** Our sliding window nonlinear least squares optimizer estimates the object pose x_t from the outputs of the frontend. Each object pose x_t is constrained by the SDF loss, frame-to-frame ICP, and pose regularization to ensure tracking remains stable.

5.6.1 Segmented visual depth

During in-hand manipulation, finger-object occlusion is inevitable and the foreground-background is ambiguous. Robust segmentation of the image stream I_t^c via prompts has successfully been demonstrated by image foundation models, like the Segment Anything Model (SAM) [Kir+23]. Trained with a vision transformer (ViT) in the data-rich natural image domain, SAM generalizes to novel scenes for state-of-the-art, zero-shot instance segmentation.

Even with SAM, in-hand object segmentation requires appropriate prompts to guide the pre-trained model. With an embodied agent, we can take advantage of robot kinematics to achieve this. Given our camera c with known projection operation Π^c, we can obtain any 3D point p as a pixel $(u, v) = \Pi^c(\mathrm{p})$ on the image I_t^c. Our insight is to use the 3D center of grasp and robot kinematics as prompts for SAM. This makes the reasonable assumption that the object exists between the robot's fingers, which is almost always the case.

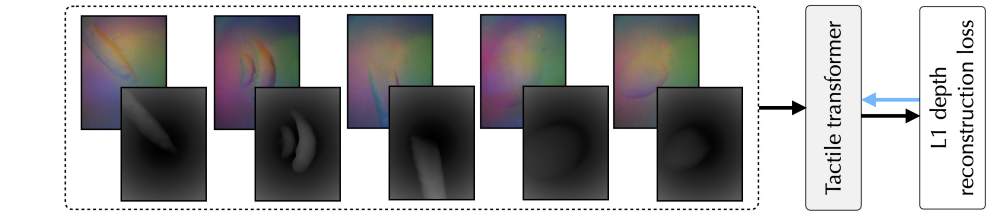
We use the 3D center of grasp, by computing the centroid of the end-effectors as a positive point prompt for SAM. However, in practice, this prompt alone doesn't suffice. First, the robot fingers frequently appear in these segmentations, which is misleading to our shape optimizer. This is solved by adding negative point prompts to fingertip pixels that we obtain by projecting the forward kinematics results. We first verify if the fingertips are unoccluded by the object, which we do by comparing against the current rendered object model. Second, SAM tends to over segment objects with distinct parts (*e.g.* different faces of the Rubik's cube). In case of these ambiguities, SAM outputs multiple masks, at different spatial scales. We apply a final pruning step to find the mask prediction closest to the average mask area we typically observe in simulation.

In Figure 5.5 (a) we show segmentation on real-world images, alongside SAM prompts. We use the ViT-L model with 308M parameters. While this achieves a speed of around 4Hz, in practice, we can use efficient segmentation models [Zha+23] for speeds up to 40Hz.

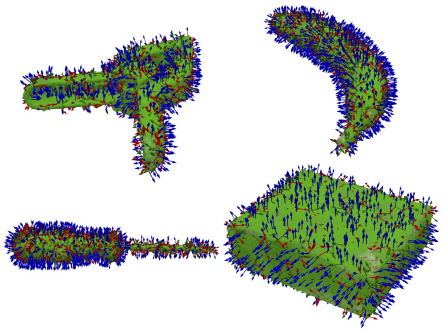
5.6.2 Tactile transformer

In contrast, vision-based touch images are out-of-distribution from images SAM is typically trained on, and does not directly provide depth either. The embedded camera perceives an illuminated gelpad, and contact depth is either obtained via photometric stereo [YDA17], or supervised learning [BCR19; Wan+21; Sod+22b; Amb+21; Sur+22b]. Existing touch-to-depth relies on convolution, however recent work has shown the benefit of a ViT for dense depth prediction [RBK21] in natural images. We train a *tactile transformer* for predicting contact depth from vision-based touch to generalize across *multiple* real-world DIGIT sensors.

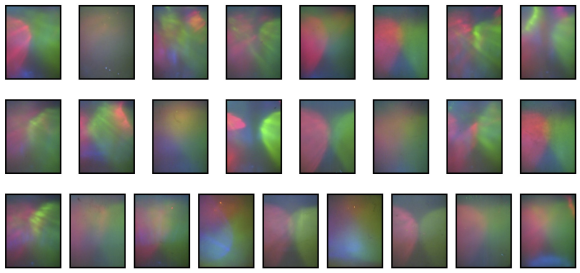
The architecture is trained entirely in tactile simulation, using weights initialized from a pre-trained image-to-depth model [RBK21]. The tactile transformer represents the inverse sensor model $: I_t^s \mapsto \hat{D}_t^s$ where $s \in \{d_{\text{index}}, d_{\text{middle}}, d_{\text{ring}}, d_{\text{thumb}}\} \in \mathcal{S}$. This architecture is based on



(a) Tactile transformer training with simulation rendered image-depth pairs



(b) Examples of posed tactile samples on YCB objects in TACTO



(c) 25 real DIGIT background images used for data augmentation

Figure 5.6: Our tactile transformer is trained in simulation with real-world augmentation. **(a)** The tactile transformer is supervised from paired RGB-depth images rendered in TACTO [Wan+22]. **(b)** Each of these samples are generated from dense, random interactions with 40 different YCB objects. **(c)** In our training, we augment the data with background images collected from 25 unique DIGIT sensors [Lam+20].

the dense vision transformer [RBK21] and is lightweight (21.7M parameters) compared to its fully-convolution counterparts [Sur+22a].

Similar to prior work [Sur+22b; Sur+22a], we generate a large corpus of tactile images and paired ground-truth depthmaps in the optical touch simulator TACTO [Wan+22]. We collect 10K random tactile interactions each on the surface of 40 unique YCB objects [Cal+17]. For sim-to-real transfer we augment the data with randomization in sensor LED lighting, indentation depth, and pixel noise. In TACTO, image realism is achieved by compositing with *template* non-contact images from real-world DIGITs. More details on data and training are below.

Model architecture. Our model architecture is based on a monocular depth network, the dense prediction transformer (DPT) [RBK21]. It comprises of a vision transformer (ViT) backbone that outputs bag-of-words features at different resolutions, finally combined into a dense prediction via a convolutional decoder. Compared to fully-convolutional methods, DPT has a global receptive field and the resulting embedding does not explicitly down-sample the image.

Training and loss metric. Our image-to-depth training dataset comprises of 10K simulated tactile interactions each on the surface of 40 YCB objects. We illustrate examples of the interactions in Figure 5.6 (b). We use the ADAM optimizer with momentum and a batch size of 100, trained with mean-square depth reconstruction loss (Figure 5.6 (a)). We start with a pre-trained small ViT [Dos+20], with an embedding dimension of 384 patch size of 16. The dataloader splits the

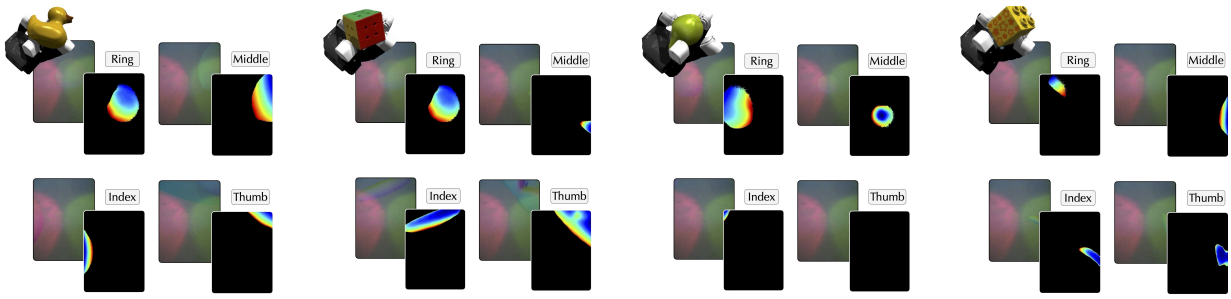


Figure 5.7: Image to depth predictions by the tactile transformer on simulated contacts. Our tactile transformer shows good performance in simulated interactions, capturing both large contact patches, as well as smaller edge features. These objects are unseen during training—as highlighted in Section 5.6.2, we demonstrate an average prediction error of 0.042 mm on simulated test images.

train, test, and validation data into 60%, 20%, and 20% respectively. We visualize additional simulation results in Figure 5.7.

Data augmentation. An important aspect of generalization and sim-to-real transfer is the augmentation applied during data collection and training. These include:

- **Real-world backgrounds.** We compose simulated renderings with real-world background images, collected from 25 different DIGIT sensors. These are shown in Figure 5.6 (c).
- **Pose variations.** Before rendering a sensor pose, we apply noise in rotation/translation and sensing normal direction. Additionally, we randomly vary the distance of penetration into the object surface.
- **Sensor lighting.** We randomize position, direction and intensity of the three DIGIT LEDs.
- **Sensor pixel noise.** We add Gaussian noise to RGB data, with a standard deviation of 7px.
- **Standard transforms.** Randomized horizontal flipping, cropping, and rotations of the tactile images.

These augmentations enable generalized performance across our multi-finger platform, where each sensor has differing image characteristics. Our tactile transformer is supervised on mean-square depth reconstruction loss against the ground-truth depthmaps from simulation. Based on the predicted depthmaps, the output is thresholded to mask out non-contact regions. The tactile transformer demonstrates an average prediction error of 0.042 mm on simulated test set. Figure 5.5 (b) shows sim-to-real performance of the tactile transformer on real-world interactions.

5.7 Backend optimizer

The backend (right column of Figure 4.1) is responsible for taking in depth and sensor poses from the frontend to build our *object model* online. This alternates between shape (Section 5.7.1) and pose optimization (Section 5.7.2) steps using samples from the visuo-tactile depth stream. Similar to other neural SLAM methods [Ort+22], the modules maintain a bank of *keyframes* over time to generate these samples.

5.7.1 Shape optimizer

For online estimation it is intractable to optimize $F^{\theta}_{\mathrm{x}_t}$ using *all* input frames as in neural radiance fields [Mil+21]. We opt for an online learning approach [Suc+21; Ort+22], which builds a subset of *keyframes* $\mathcal{K}$ on-the-fly to optimize over. The backend must both **(i)** accept new keyframes based on a criteria, and **(ii)** replay old keyframes in the optimization to prevent *catastrophic forgetting* [Suc+21]. Each iteration of the shape optimizer replays a batch $k_t \in \mathcal{K}$ of size 10 per sensor to optimize our network. This includes the latest two frames, and a weighted random sampling of past keyframes based on average rendering loss.

The initial visuo-tactile frame is automatically added as a keyframe $\mathcal{K}_0 = \{\hat{D}^s_0 \mid s \in \mathcal{S}\}$, and every subsequent keyframe $\mathcal{K}_t$ is accepted using an information gain metric [Suc+21]. For this, the average rendering loss is computed from the frozen network $F^{\theta}_{\mathrm{x}_t}$ using the given keyframe pose and compared against a threshold $d_{\text{thresh}} = 0.01\,\text{m}$. Finally, if we have not added a keyframe for an interval $t_{\text{max}} = 0.2\,\text{secs}$, we force one to be added.

Sampling and SDF loss. At each iteration, we sample coordinates in the neural volume from k_t to optimize the neural weights θ. The first step is to sample a batch of pixels u_{k_t} from k_t—a mix of surface and free-space pixels. While surface pixels directly supervise the SDF zero level-set, free-space pixels carve out the neural volume. In our implementation, we sample 50% of camera pixels in free-space, while we only sample surface pixels for touch. Through each pixel $u \in \mathrm{u}_{k_t}$ given their corresponding sensor pose, we project a ray into the neural volume. Similar to Ortiz et al. [Ort+22], we sample P_u points per ray, a mix of stratified and surface points.

With these samples, we compute an SDF prediction $\widehat{\mathrm{d}}_u$ for each $\hat{D}_t \in k_t$, as the batch distance bound [Ort+22]. For each ray, we split the samples into P^{f}_u and P^{tr}_u based on $\widehat{\mathrm{d}}_u$ lies within the truncation distance $d_{\text{tr}} = 5\,\text{mm}$ from the surface. Our shape loss $\mathcal{L}_{\text{shape}} = \mathcal{L}_{\text{f}} + w_{\text{tr}}\mathcal{L}_{\text{tr}}$, with $w_{\text{tr}} = 10$, resembles the truncated SDF loss of Azinović et al. [Azi+22]:

$$\mathcal{L}_{\text{f}} = \frac{1}{|\mathrm{u}_{k_t}|}\sum_{u\in\mathrm{u}_{k_t}}\frac{1}{|P^{\text{f}}_u|}|F^{\theta}_{\mathrm{x}_t}(P^{\text{f}}_u) - d_{\text{tr}}| \qquad \text{and} \qquad \mathcal{L}_{\text{tr}} = \frac{1}{|\mathrm{u}_{k_t}|}\sum_{u\in\mathrm{u}_{k_t}}\frac{1}{|P^{\text{tr}}_u|}|F^{\theta}_{\mathrm{x}_t}(P^{\text{tr}}_u) - \widehat{\mathrm{d}}_u|$$

As a proof-of-concept, we perform a single-touch optimization using data from TACTO [Wan+22]. Given the input tactile image shown in Figure 5.8, we compute the depth image and sample points in the neural volume. Upon running a few hundred iterations of SDF optimization, we can converge to an implicit surface that matches well with the ground-truth. The rendered depth and normals also resemble that of the ground-truth. We then extend this method over a trajectory of tactile sliding data from Figure 4.6, to reconstruct the implicit surface shown in Figure 5.9.

Implementation details. The neural field is optimized via Adam [KB14] with learning rate of $2\text{e-}4$ and weight decay of $1\text{e-}6$. Instant-NGP uses a hash table of size 2^{19} for positional encoding, followed by a 3-layer MLP with 64 dimensional width. We use a uniform random weights θ_{init}

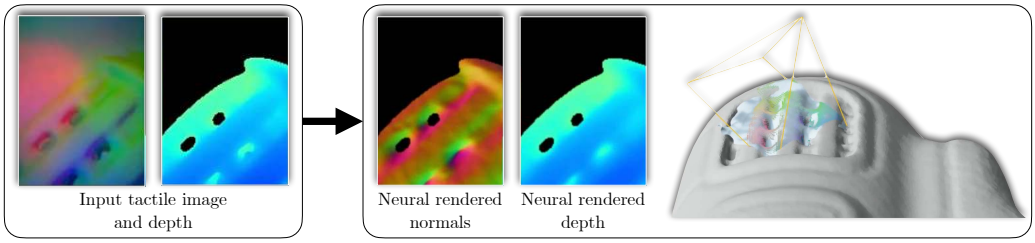


Figure 5.8: Single frame SDF optimization results for a single touch. Given an input tactile image **[left]**, we first extract its depthmap and contact mask with our tactile depth network (Section 4.3.1). Then, we optimize the neural field with a truncated-SDF loss from back-projecting sampled rays. Finally, we apply marching cubes to extract the zero level-set, and visualize the 3D surface superimposed over the ground truth mesh **[right]**. Also highlighted are the rendered predictions of normals and depth from the neural field, which match well with the ground-truth mesh.

and initialize the SDF by running 500 shape iterations using the first keyframe $\mathcal{K}_0$.

For evaluating the neural field we freeze the network and query a 200^3 feature grid. The feature grid's extents are defined as a bounding box of 15 cm side, centered at the object's initial pose x_0. When training, we apply a series of bounding-box checks post hoc, to eliminate any ray samples P_u found outside this bounding box. Mesh visualizations (Figure 5.13) are periodically generated via marching-cubes on the feature grid. We add color to the mesh by averaging the colored object pointcloud with a Gaussian kernel.

5.7.2 Pose optimizer

Before each shape iteration, we use a pose graph [DK17] to refine the object pose x_t with respect to the frozen neural field $F^{\bar{\theta}}_{\mathrm{x}_t}$. We achieve this by *inverting* the problem to instead optimize for the 6-DoF poses in a sliding window of size n. At timestep t, if we have accumulated N keyframes,

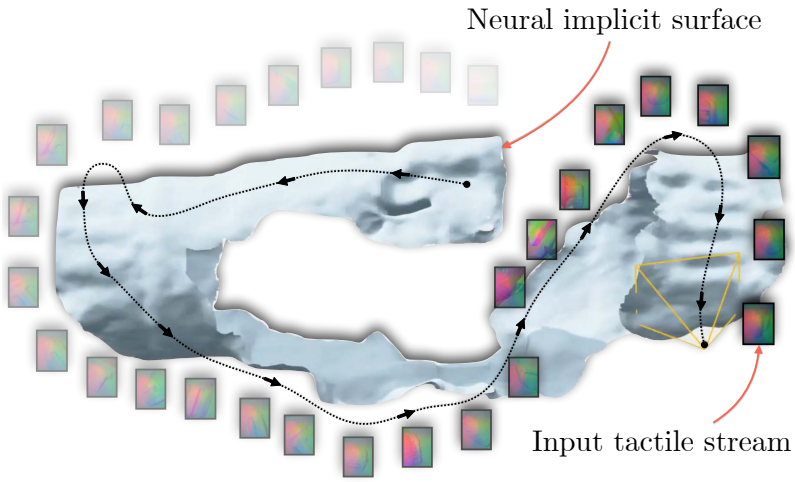


Figure 5.9: Online optimization of the neural field over the tactile image sequence, to give us surface of the `power_drill` object. At each timestep, the reconstruction loss is optimized using samples from a subset of tactile keyframes. We see that our method can obtain accurate reconstructions of the object's local surface from a posed tactile image stream.

this is represents poses $\mathcal{X}_t = (\mathrm{x}_i)_{N-n \leq i \leq N}$ and measurements $\mathcal{M}_t = \left(\hat{D}_i^s \mid s \in \mathcal{S}\right)_{N-n \leq i \leq N}$. Similar to pose updates in visual SLAM [Yen+21; Suc+21; Zhu+22], the network weights $\bar{\theta}$ are frozen and we estimate the $SE(3)$ poses $\mathcal{X}_t$ instead.

We formulate the problem as a nonlinear least squares optimization with custom measurement factors in Theseus [Pin+22]. While prior work uses gradient descent [Yen+21], we instead use a second-order LevenbergMarquardt (LM) solver, which provides faster convergence [DK17]. The pose graph, illustrated in Figure 5.5 (c), solves for:

$$\hat{\mathcal{X}}_t = \underset{\mathcal{X}_t}{\operatorname{argmin}} \mathcal{L}_{\text{pose}}(\mathcal{X}_t \mid \mathcal{M}_t, \bar{\theta}) \qquad \text{where} \qquad \mathcal{L}_{\text{pose}} = w_{\text{sdf}}\mathcal{L}_{\text{sdf}} + w_{\text{reg}}\mathcal{L}_{\text{reg}} + w_{\text{icp}}\mathcal{L}_{\text{icp}}$$

- **SDF loss $\mathcal{L}_{\text{sdf}}$.** We use the shape loss $\mathcal{L}_{\text{shape}}$, modified such that we sample only about surface points of each ray. This works well for both visual and tactile sensing as we have higher confidence in SDFs about the surface of the object than in free-space. For each depth measurement in $\mathcal{M}_t$, we sample surface points over M rays, and average the SDF loss along each ray. This results in an $M \times n$ SDF loss, which we use to update the se(3) lie algebra of $\mathcal{X}_t$. We implement a custom Jacobian for this cost function, which is up to $4\times$ more efficient than PyTorch automatic differentiation.
- **Pose regularizer $\mathcal{L}_{\text{reg}}$.** We apply a weak regularizer between consecutive keyframe poses in $\mathcal{X}_t$ to ensure the relative pose updates stay well-behaved. This is important for robustness to noisy frontend depth and incorrect segmentations.
- **ICP loss $\mathcal{L}_{\text{icp}}$.** We further apply iterative closest point (ICP) between the current visuotactile pointcloud $\Pi^{-1}(\mathcal{M}_t)$ and previous pointcloud $\Pi^{-1}(\mathcal{M}_{t-1})$. This gives us frame-to-frame constraints in addition to the frame-to-model $\mathcal{L}_{\text{sdf}}$.

We use the vectorized $SE(3)$ pose graph optimizer in Theseus [Pin+22], with 20 LM iterations of step size 1.0. The keyframe window size $n = 3$ and we run 2 pose iterations for each shape iteration. The weighting factors for each loss are $w_{\text{sdf}} = 0.01$, $w_{\text{reg}} = 0.01$, and $w_{\text{icp}} = 1.0$.

5.8 Results

Our multi-fingered robot hand is presented with a novel object, placed randomly between its fingertips. It rotates the object in-hand, through a proprioception-driven policy [Qi+22], which gives rise to a stream of visual and tactile signals. We combine the visual, tactile, and proprioceptive sensing into our online neural field, for a persistent, evolving 3D representation of the unknown object. The full pipeline of our NeuralFeels perception stack is illustrated in Figure 5.3. We also summarize our experiments and findings in our webpage.

We evaluate NeuralFeels over simulated and real-world interactions, totaling up to 70 experiments over different object classes. Details of the dataset can be found in Section 5.3.2. First, we

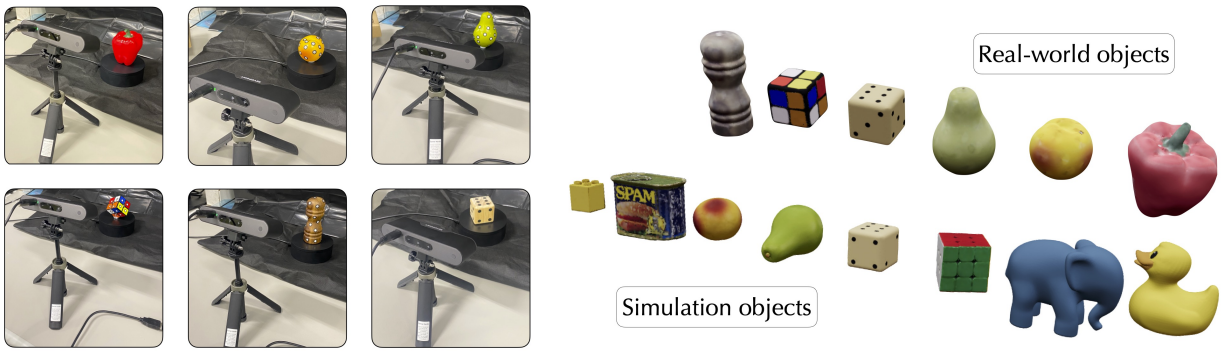
demonstrate SLAM results for novel objects, and highlight some qualitative examples. Next, we demonstrate pose-tracking when we have a priori shape of the manipuland. Finally, we analyze the role touch plays in improving perception under occlusion and visual sensing noise.

5.8.1 Metrics and baseline

Pose and reconstruction metrics. We use the symmetric average Euclidean distance metric (ADD-S) to evaluate the pose tracking error over time [Tre+23]. The ADD metric is commonly used in manipulation [Xia+17; BCR19; Tre+18; Tre+23] as a geometrically-interpretable distance metric for pose error. It is computed by sub-sampling the ground-truth object mesh and averaging the Euclidean distance between the point-set in the estimated and ground-truth object pose frames. Rather than pairwise distance, ADD-S considers the closest point distance, which disambiguates symmetric objects.

For reconstruction, we compare how *accurate* (precision) and *complete* (recall) the neural SDF is in comparison to the ground-truth mesh. The F-score, an established metric in the multi-view reconstruction community [Kna+17; Tat+19], combines these two criteria into an interpretable $[0-1]$ value. To compute this, we first sub-sample the ground-truth and reconstructed meshes, and transform both to the common object-centric reference frame. Given a distance threshold, in our case $\tau = 5$ mm, *precision* measures the percentage of reconstructed points within τ distance from the ground-truth points. Conversely, *recall* measures the percentage of ground-truth points within τ distance from the reconstructed points. The harmonic mean of these two quantities give us the F-score, which captures both surface reconstruction accuracy and shape completion. Broadly, a higher F-score with tighter τ bounds implies better object reconstructions. For brevity, we refer to ADD-S and F-score as the *pose metric* and *shape metric* respectively.

Ground-truth shape. For each object, the ground-truth shape is obtained from offline scans (Figure 5.10). For this, we use a commercial dual-camera infrared scanner, the Revopoint POP



(a) Ground-truth object scanning **(b)** Resulting ground-truth meshes

Figure 5.10: Object ground-truth with dual-camera infrared scanner. (a) Objects are placed on a turntable and scanned, followed by post-processing to ensure complete, accurate meshes. **(b)** Meshes visualized for the real and simulated FeelSight objects.

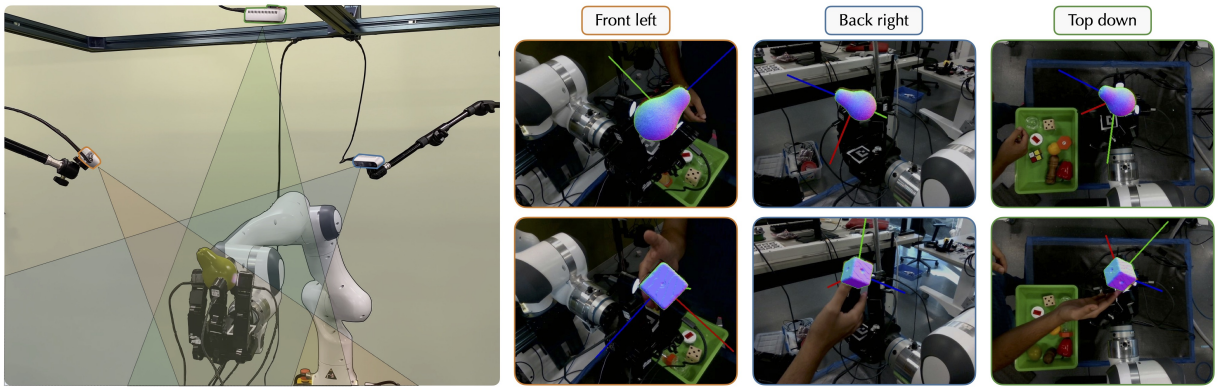


Figure 5.11: Robot cell for pseudo-ground-truth tracking. Each of the three camera's captures a different field-of-view of the interaction **(left)**. For a pseudo-ground-truth, we pass the RGB-D stream from all three cameras into our pipeline, with known shape obtained from scanning. The output pose tracking **(right)** represents the ground-truth we compare to in the real-world results.

3 [Rev23]. The hardware can scan objects from a close range with a minimum precision of 0.05 mm. Each real-world object is placed on a turntable and scanned while rotating about its axis (Figure 5.10 (a)). For object's that lack texture, an artificial dot pattern is tracked by adding stickers. After generating the scans, we perform hole-filling for unseen regions of the object, like the bottom. Figure 5.10 (b) shows all the scanned meshes—a few meshes are directly sourced from the YCB [Cal+17] and ContactDB [Bra+19] datasets.

Pseudo ground-truth pose tracking. Ground-truth object pose is straightforward in simulation experiments, directly exposed by IsaacGym [Mak+21]. In the real-world, we estimate a pseudo ground-truth, via multi-camera pose tracking of the experiment. Instrumented solutions, such as 3D motion capture, are infeasible as it both visually and physically interferes with the experiments. We opt to install two additional cameras (Section 5.3.2) and run NeuralFeels in pose tracking mode with the ground-truth object shape. We pass three RGB-D cameras as input into our pose tracking pipeline to use as a pseudo ground-truth estimate. This consists of three unique cameras (*front left, back right, top down*) with complementary but overlapping fields-of-view (Figure 5.11 and Figure 5.17 (b)). With this broad perspective of the scene, known shape from ground-truth scans, and the tracker running at 0.5Hz, we can obtain an accurate estimation of object pose at each timestep. This represents the *best tracking estimates given known shape and occlusion-free vision*. In the real-world,

Compute and timings. All results are generated from playing-back the trials at a publishing rate of 1 Hz. Experimentally, however, we can run the pose optimizer at 10 Hz and full backend at 5 Hz. Figure 5.17 (a) has a minimal robot setup of an online SLAM system with rotation policy in-the-loop. Experiments are run on an Nvidia GeForce RTX 4090, while the aggregate results are evaluated on a cluster with Nvidia Tesla V100s.

5.8.2 Neural SLAM: object pose and shape estimation

In this section, we evaluate NeuralFeels' ability for embodied spatial reasoning from scratch. We present the robot with a novel object, and the robot is tasked with building an object model on-the-fly. This is typical where robots continually learn from interaction, such as when deployed in unstructured household environments. We make no assumptions about the object geometry, which is built from scratch, or manipulation actions, which are decided at deployment. We process visuo-tactile data sequentially with no access to future information or category-level priors. This formulation aligns with other dexterous manipulation work [Han+22; Qi+22; Qi+23; Che+23a], and is less restrictive than that of FingerSLAM [ZBA23], where the object is always in contact with a single tactile sensor and the camera is unobstructed.

We evaluate over a combined 70 experiments in simulation and real-world across of 14 different objects. The objects are placed in-hand, after which the policy collects 30 seconds of vision, touch, and proprioception data. As each run is non-deterministic, we average our results across 5 different seeds, resulting in a total of 350 trials. The first frame of each sequence only presents limited visual knowledge: a single side of *Rubik's cube* or *large dice*; the underside of the *rubber duck*. Through the course of any 30 second sequence, in-hand rotation exposes previously unseen geometries to vision and touch fills in the rest of the occluded surfaces. In Figure 5.12, we present the main set of results, where we compare the multimodal fusion schemes against ground-truth.

Object reconstructions. Figure 5.12 (a) shows the final shape metric at the end of each sequence for a fixed threshold τ. Here we pick $\tau = 5$ mm for this evaluation, around 3% of the maximum diagonal length of the objects. Greater the value of the shape metric, the closer the surface reconstructions are to ground-truth. We observe large gains when incorporating touch, with surface reconstructions on average 15.3% better in simulation and 14.6% better in the real-world. Our final reconstructions, as seen in Figure 5.12 (e), have a median error of 2.1mm in simulation and 3.9mm in the real-world. Additionally, the second plot compares the final shape metrics against a range of τ thresholds. Here we observe that multimodal fusion leads to consistently better shape metrics across all τ values in simulation and the real-world.

Object pose drift. As SLAM is the exemplar of a chicken and egg problem, there is a strong correlation between a low shape metric and high pose metric. Empirically, we observe larger pose drift in the initial few seconds due to incomplete geometry, which levels off with further exploration. For fair comparisons we initialize the object's canonical pose to the ground-truth, but this is not necessary otherwise. With this initialization, we ignore the pose metric over the first five seconds, as it is ill-defined.

Figure 5.12 (b) plots the drift of the object's estimated pose with respect to the ground-truth, lower being more accurate. We observe better tracking with respect to the vision-only baseline, with improvements of 21.3% in simulation and 26.6% in the real-world. Table 1 in Figure 5.12 (c) reports the number of failures in vision-only tracking compared to NeuralFeels. Here, a failed

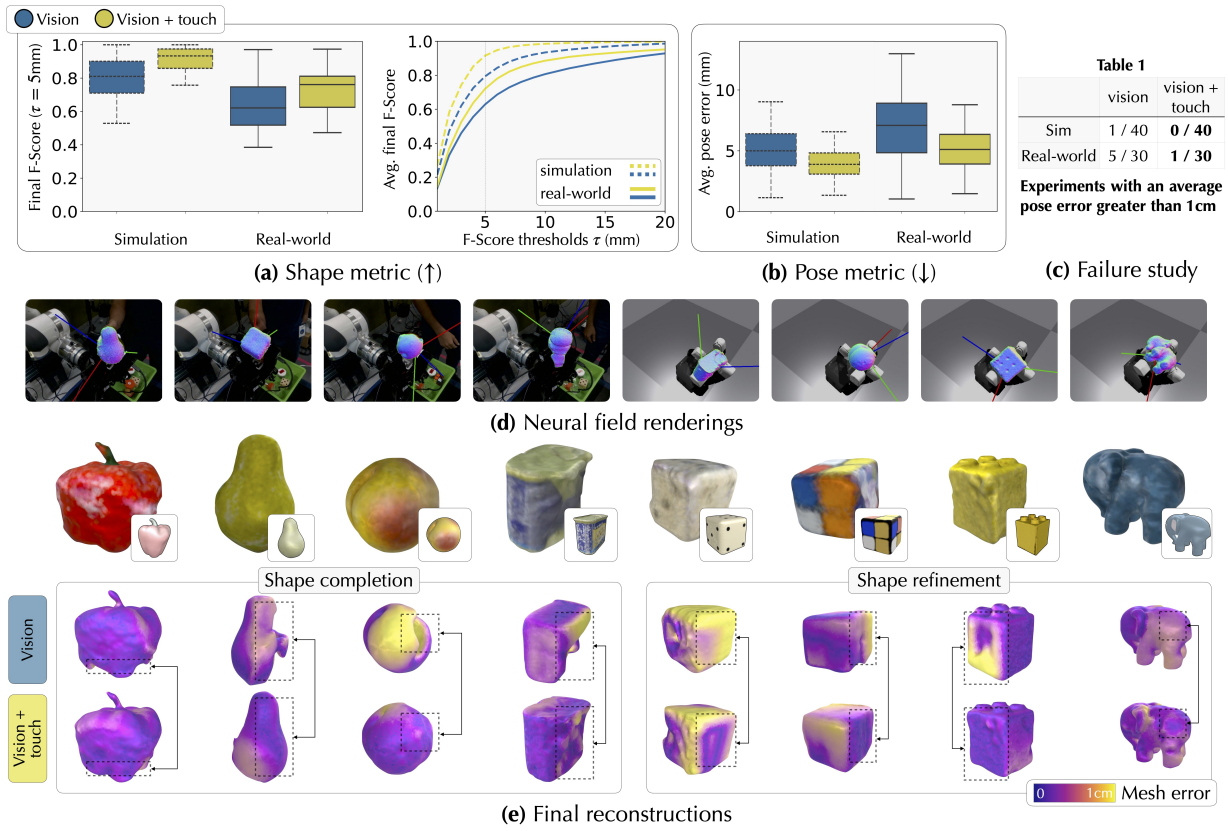


Figure 5.12: Summary of SLAM experiments. **(a, b)** We present aggregated statistics for SLAM over a combined 70 experiments (40 in simulation and 30 in the real-world), with each trial run over 5 different seeds. We compare across simulation and real-world to show low pose drift and high reconstruction accuracy. **(c)** Table 1 illustrates the number of trials that our method fails to track (and reconstruct) the object. **(d)** Representative examples of the final object pose and neural field renderings from the experiments. **(e)** The final 3D objects generated by marching cubes on our neural field. Here, we highlight the role tactile plays in both shape completion and shape refinement.

experiment is defined as when the average pose drift exceeds an empirical threshold of 10 mm.

Qualitative results. Figure 5.12 (d) visualizes the rendered normals of the posed neural field at the end of each experiment, with the 3D coordinate axes superimposed. The final 3D reconstructions, generated via marching cubes, are shown in Figure 5.12 (e) alongside the ground-truth meshes. Below that, we highlight the gains with visuo-tactile integration, with examples of shape completion and refinements.

In Figure 5.13 we show the incremental pose tracking and reconstructions of objects across different time slices of a few representative experiments. We present two results from the real-world, *bell pepper* and *large dice*, and two from simulation, *rubber duck* and *peach*. At each timestep, we highlight the input stream, frontend depth and output object model. The 3D visualizations are generated by marching-cubes, in addition to the rendered normals of the neural field projected onto the visual image. In each case, we partially reconstruct the object at the initial frame, and build the surfaces out progressively over time.

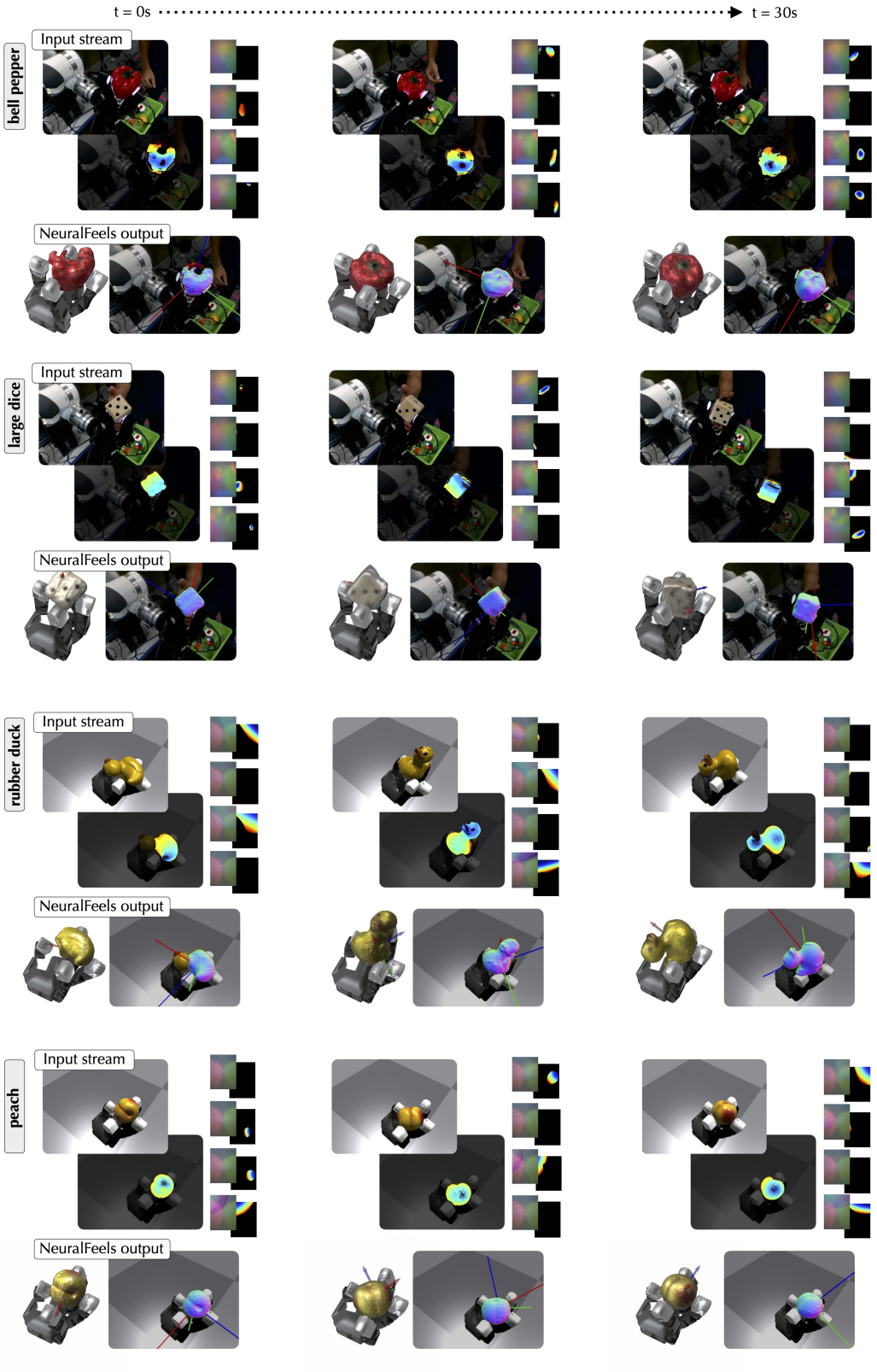


Figure 5.13: Representative SLAM results. In both real-world and simulation, we build an evolving neural SDF that integrates vision and touch while simultaneously tracking the object. We illustrate the input stream of RGB-D and tactile images, paired with the posed reconstruction at that timestep.

5.8.3 Neural tracking: object pose estimation given shape

As a special case of NeuralFeels, we demonstrate superior multimodal pose tracking when provided the CAD models of the objects at runtime. Tracking known geometries is an active area of research in visual SLAM [Han+22; Lab+22], with some work that incorporates touch as well [YR18; Lam+19; Sod+21; Sur+22a; Bau+23]. This is applicable in environments like warehouses and manufacturing lines, where robots have intimate knowledge of the manipulands [Bau+23]. It is further useful in household scenarios, where the robot has already generated an object model through interaction.

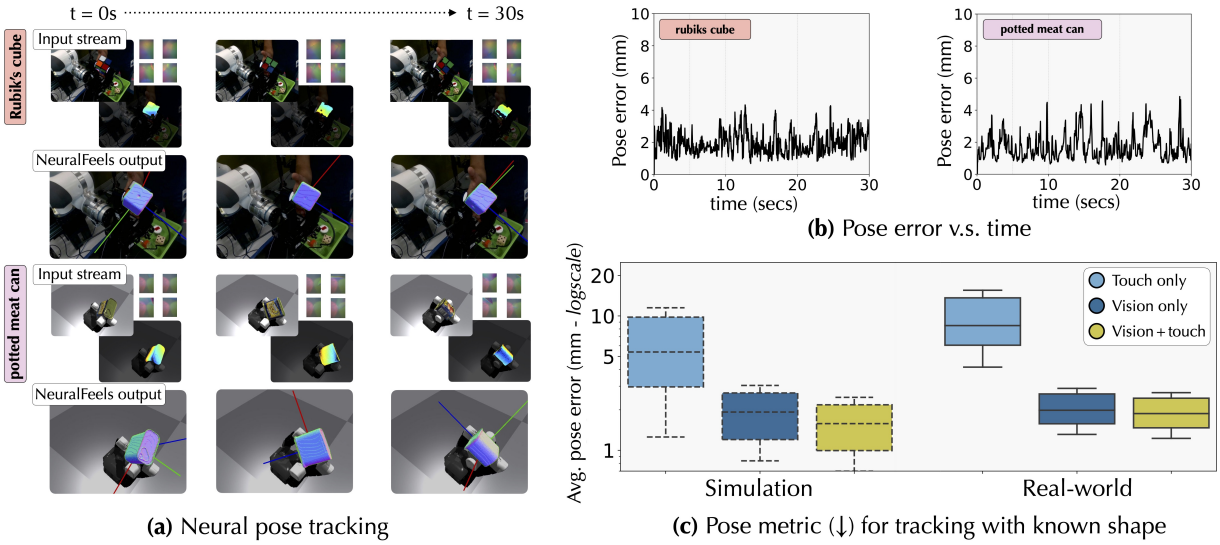
In implementation, the object's SDF is pre-computed from a given CAD model. During runtime, we freeze the weights of the neural field, and only perform visuo-tactile tracking with the frontend estimates. Similar to the SLAM experiments, we run each of the 70 experiments over 5 seeds, and report the pose metrics with respect to ground-truth.

Results from pose tracking. Figure 5.14 (a) shows some qualitative examples of tracking the pose of the *Rubik's cube* and *potted meat can* with vision and touch. For the given examples, the pose metric over the sequences are plotted in Figure 5.14 (b). We observe low, bounded pose error even with imprecise visual segmentation and sparse touch signals. In Figure 5.14 (c) we observe the role touch plays in reducing the average pose error over all experiments to the range of 2.3 mm. Given the CAD model, we observe that incorporating touch can refine our pose estimates, with a decrease in average pose error by 22.29% in simulation and 3.9% in the real-world. As addressed in Section 5.9, the less-pronounced contacts in the real-world can explain this disparity. In the following section, we highlight greater improvements with respect to the baseline when visual sensing is suboptimal.

5.8.4 Perceiving under duress: occlusion and visual depth noise

In this section, we explore the broader benefits of fusing touch and vision through ablations on visual sensing properties. The previous results were achieved through the iterative co-design of perception and hardware, such that we have favorable camera positioning and precise stereo depth tuning. Indeed, this attention to detail is necessary for practitioners [Han+22; Che+23a], *but can we also harness touch to improve over sub-optimal visual data?* We consider two such scenarios in simulation, where we can freely control these parameters, and evaluate on the pose tracking problem from the previous section.

The effects of camera-robot occlusion. In an embodied problem, third-person and egocentric cameras are both susceptible to occlusion from robot motion and environment changes. For example, if we were to retrieve a cup off the top shelf in the kitchen, we rely primarily on tactile signals to complete the task. For the perception system, this translates to the object of interest disappearing from the field of view, while local touch sensing is still unaffected. To emulate this we consider tracking the pose of a known *Rubik's cube*. We simulate 200 different cameras in a



(a) Neural pose tracking

(b) Pose error v.s. time

(c) Pose metric (↓) for tracking with known shape

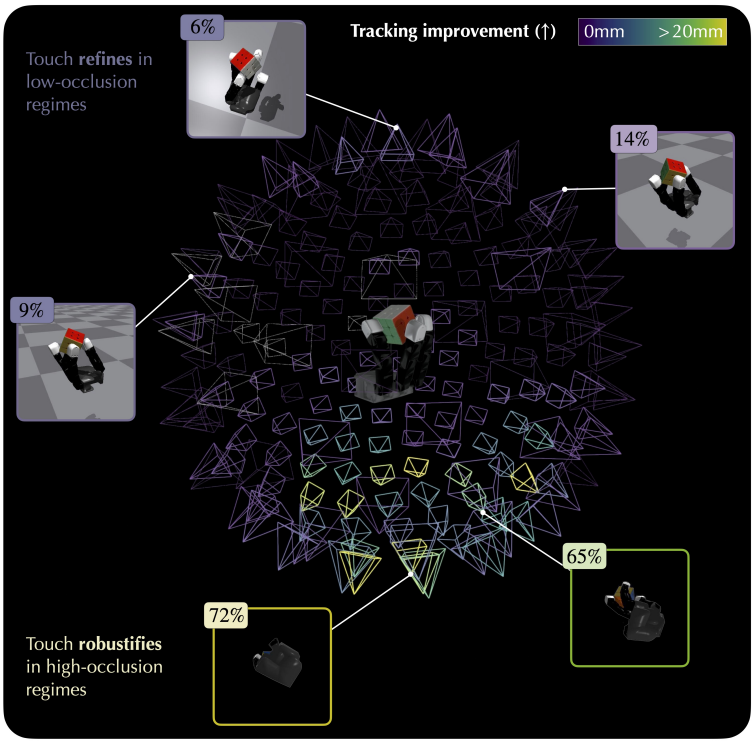
Figure 5.14: Neural pose tracking of known objects. (a) With known ground-truth shape, we can robustly track objects such as the Rubik's cube and potted meat can. **(b)** We observe reliable tracking performance, with average pose errors of 2 mm through the sequence. **(c)** With a known object model and good visibility, touch plays the role of pose refinement.

sphere of radius 0.5 m, each facing towards the robot. As shown in Figure 5.15 (a), each camera captures a unique vantage point of the same in-hand sequence, with varying levels of robot-object occlusion. This serves as proxy for occlusion faced by an egocentric or fixed camera when either the hand or environment occludes the object.

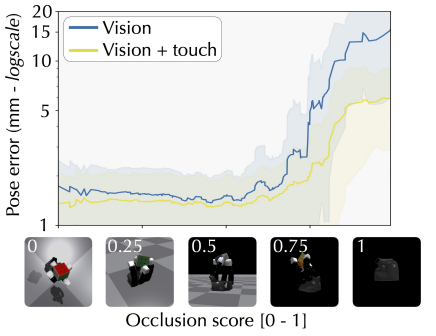
To simplify the experiment, we assume the upper-bound performance of the vison-only frontend by providing ground-truth object segmentation masks. We characterize the visibility in terms of an *occlusion score* by calculating the average segmentation mask area for each viewpoint, and normalizing them to $[0-1]$. For example, scores closer to 0 correspond to viewpoints beneath the hand (most occluded), while those closer to 1 correspond to cameras placed atop (least occluded). We run pose tracking experiments for each of the 200 cameras in two modes: vision-only and visuo-tactile and compare between them.

In Figure 5.15 (a) we colormap each camera view based on the pose tracking improvements from incorporating touch. On average the improvement across all cameras is 21.2%, and it peaks at 94.1% at heavily occluded views. We inset frames from a few representative viewpoints and their corresponding relative improvement with visuo-tactile fusion. In Figure 5.15 (b) the pose error for each modality is further plotted versus the $[0-1]$ occlusion score. This corroborates the idea that touch refines perception in low-occlusion regimes and robustifies it in high-occlusion regimes.

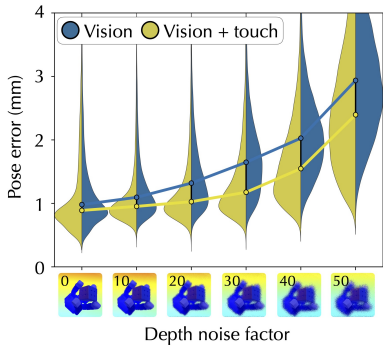
The effects of noisy visual depth. Depth from commodity RGB-D sensors are degraded as a function of camera-robot distance, environment lighting, and object specularity. Even in ideal scenarios, the RealSense depth algorithm has 35 hyperparameters [Kes+23] that considerably af-



(a) Object occlusion: Tracking improvement by fusing touch (↑)



(b) Pose error with object occlusion (↓)



(c) Pose error with depth noise (↓)

Figure 5.15: Ablations on occlusions and sensing noise. (a) With occluded viewpoints, visuo-tactile fusion helps improve tracking performance with an unobstructed local perspective. We quantify these gains across a sphere of camera viewpoint to show improvements, particularly in occlusion-heavy points-of-view. **(b)** We observe that touch plays a larger role when vision is heavily occluded, and a refinement role when we there is negligible occlusion. **(c)** With larger noise in visual depth, tactile help curb large pose tracking errors.

fect the frontend input to NeuralFeels. To simulate this, we corrupt the depth maps progressively with a realistic RGB-D noise, and observe the tracking performance for a known geometry.

As implemented by Handa et al. [Han+14], we simulate common sources of depth-map errors as a sequence of pixel shuffling, quantization, and high frequency noise. The depth noise factor D determines the magnitude of these operations, with the depth-maps visualized in Figure 5.15 (c). While all prior simulation experiments have been collected with $D=5$, here we vary the magnitude from $0-50$ in intervals of 10. At each noise level, we run pose tracking across the 5 Rubik's cube experiments with 5 unique seeds, resulting in a total of 150 experiments. In Figure 5.15 (c) we plot error against the noise factor D, showing an expected upward trend in error with noise. However, we see markedly better tracking when fusing touch, especially in high-noise regimes.

5.9 Discussion

NeuralFeels achieves robust object-centric SLAM through interaction. To the best of our knowledge, NeuralFeels is the first demonstration of full-SLAM for multimodal, multifinger

manipulation. We are inspired by computer vision systems that achieve high-fidelity neural reconstructions without pose annotation [Suc+21; Zhu+22; Wen+23] through online learning. They highlight the benefit of co-designed pose tracking and reconstruction, which has also shown promise in manipulation systems [Sur+21; ZBA23]. More broadly, our stack relies on recent progress in somewhat disparate fields: SLAM, neural rendering, tactile sensing, and reinforcement-learning.

As shown in the Figure 5.12 (a), we achieve average reconstruction F-scores of 81% across simulation and real-world experiments on novel objects. Simultaneously, we stably track these objects amidst interaction with minimal drift, an average of 4.7 mm. While the vision-only baseline may suffice for some scenarios, the results validate the utility of rich, multimodal sensing for interactive tasks. This corroborates years of research in interactive perception from touch and vision [Smi+21; Sur+22b; Bau+23], now applied on dexterous manipulation platforms.

Touch and proprioception ground embodied perception. Interactive perception is far from ideal, an embodiment can more often than not get in the way of sensing. As seen in Figure 5.13, in-hand manipulation suffers from challenges such as frequent occlusions, limited field-of-view, noisy segmentation, and rapid object motion. To tackle this, proprioception helps focus the perception problem: we can accurately singulate the object of interest through embodied prompting (Section 5.6.1). When combined with touch, we robustify our visual estimates by giving us a window into local interactions. These are evident in simulated / real SLAM and pose tracking experiments, where multimodal fusion leads to improvements of 15.3% / 14.6% in reconstruction and 21.3% / 26.6% in pose tracking.

Qualitatively, we see touch performs two key functions: **(i)** disambiguating noisy frontend estimates and **(ii)** providing context in the presence of occlusion. The former alleviates the effect of noisy visual segmentation and depth with co-located local information for mapping and localization. The latter provides important context hidden from visual sensing, like the occluded face of the *large dice* or back of the *rubber duck*. The final reconstructions in Figure 5.12 (e) support these findings, with improved shape completion and refinement. This is important in the few-shot interactions of everyday life, where the richer sensing can create better object models.

The largest gains from incorporating touch are in heavy-occlusion regimes (Figure 5.15 (a)), where we can observe up to 94.1% improvements at certain camera viewpoints. To our knowledge, this co-design of perception and hardware has not been explored by practitioners before. This doesn't just demonstrate the complementary nature of the modalities, but further, the ideal configurations for occlusion-free manipulation. Finally, our results in tactile-only tracking (Figure 5.14 (c)) support the analysis of Smith et al. [Smi+20] that learning exclusively from touch leads to poor performance as it lacks any global context.

Modularity marries pre-training with online learning. As opposed to an end-to-end perception, NeuralFeels is fully interpretable due to its modular construction. This allows us to combine

foundational models trained on large-scale image and tactile data (*frontend*), with SLAM as online learning (*backend*). Furthermore, our backend is a combination of state-of-the-art neural models [Mül+22] with classical least-square optimization [Pin+22] that have found success in SLAM [Cad+16]. Chaining these systems together, we can achieve first-of-its-kind multimodal SLAM results without explicit training in the domain. This is crucial given the dearth of training data for in-hand tasks, and robot manipulation in general.

This modular design has benefits for future generalization of our system: **(i)** Other models of tactile sensors [YDA17; Wan+21; Als+19] can be easily integrated as long as they can be accurately simulated; **(ii)** alternate scene representations [Bar+21; Ker+23] can supplant our neural field model, as required; **(iii)** additional state knowledge can be seamlessly integrated as factor graph cost functions, *e.g.* tactile odometry [ZBA23] and force-constraints [Sur+21]; **(iv)** any combination of tactile and visual sensors can be fused into our multimodal framework with appropriate calibration and kinematics.

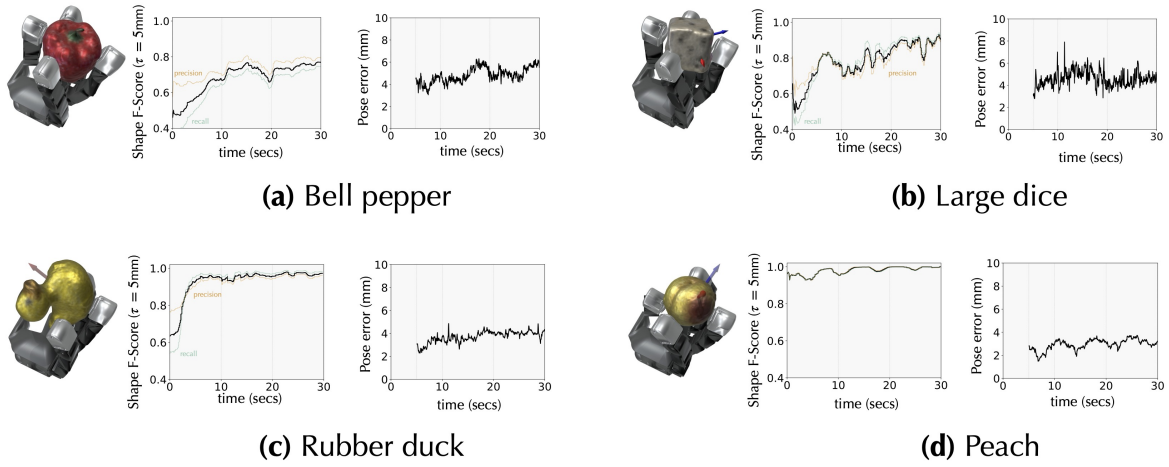
5.9.1 System limitations

We present some of the limitations and promising directions for future work:

- **Generic 3D priors for object reconstruction.** For each experiment with a novel object, our method learns a 3D geometry *from scratch* to best explain the visuo-tactile sensor stream. The pose tracker has a higher chance of failure in the initial few seconds, when the neural SDF is a poor-approximation of the full object due to limited sensor coverage. We further note that our rotation policy might not completely explore the object in the real-world, resulting in a lower average final F-Score of 81%. Out-of-scope in our work, but of great interest in the visual learning community [Par+19; Wu+23; Hon+23], is leveraging pre-trained models for an initial object prior. Given an initial occluded view, careful integration of these large reconstruction models trained via category [Par+19] or multi-view supervision [Wu+23; Hon+23] may yield an initial-guess SDF that we refine over time with vision and touch. In manipulation, Wang et al. [Wan+18] have seen promising results in using shape priors for visuo-tactile reconstruction of fixed objects.
- **Sim-to-real adaptation.** Our findings indicate that while multimodal fusion performs well both in simulation and the real-world, its benefits are less pronounced in real-world deployment. This is a common problem in sim-to-real applications, and we qualitatively identify several domain gaps that explain this: **(i)** the DIGIT elastomer is less sensitive in real-world deployment, leading to sparser contact predictions; **(ii)** our RL policy is less stable in the real-world (sometimes requiring human intervention) and causes rapid jumps in object motion; **(iii)** noise in proprioception is only indirectly modelled as uncertainty terms in estimation. To tackle these, we must leverage work in sim-to-real generalization for tactile simulation [HBM23] and reinforcement-learning [Qi+23].
- **System design considerations.** We identify viable engineering improvements that can

be made towards a general-purpose system. We are currently restricted to a fixed-camera setup, with an online hand-eye calibration or egocentric vision, this can be relaxed. Depth uncertainty [DD23] is valuable information for our neural model to handle visually-adversarial objects like glass and metal. To achieve true real-time frequencies, efficiency gains can be made in the pose optimizer and frontend estimation. Finally, we can increase robustness by using the color information for feature-based tracking of objects [DMR18].

5.10 Additional results



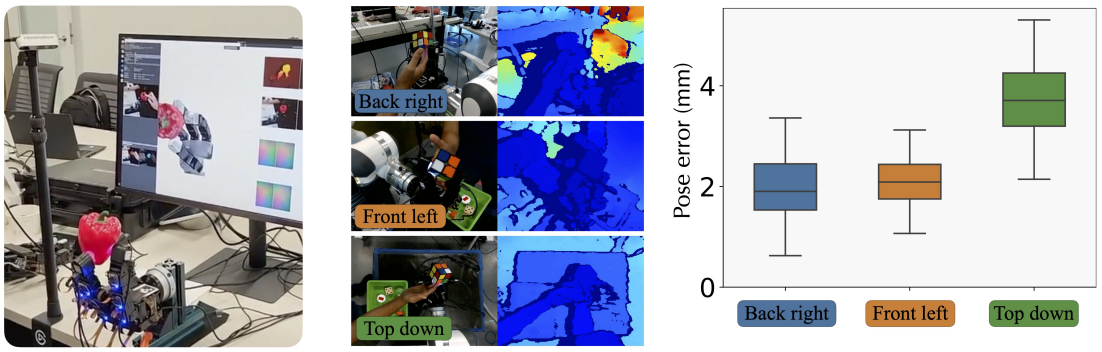
(a) Bell pepper **(b)** Large dice

(c) Rubber duck **(d)** Peach

Figure 5.16: Shape and pose metrics over time for in-hand SLAM. Here, we plot the time-varying metrics for experiments visualized in Figure 5.13. First, we note the gradual increase in F-score over time with further coverage. Additionally, we have bounded pose drift over time—for each experiment we omit the first five seconds as the metric is ill-defined then.

Shape and pose metrics over time. In Figure 5.16, we plot these metrics for each of the experiments in Figure 5.13, instead against $0-30$ sec timesteps. For shape, we observe gradual convergence to an asymptote close to 1.0, indicating evolution of both shape completion and refinement over time. Also visualized here is the precision and recall metrics over time, whose harmonic mean represents the F-score. For pose, we observe stable drift over time, indicating the estimated object pose lies close to the ground-truth estimate.

Effect of camera viewpoint in the real-world. In Section 5.8.4, we establish the relationship between occlusion/sensing noise and pose error. Here, we run additional experiments, on a limited set of viewpoints in the real-world. Figure 5.17 (b) shows the RGB-D data from three cameras *front left, back right, top down*, at distances of 27 cm, 28 cm, and 49 cm respectively from the robot. We run our vision-only pose tracker with known shape using each of three cameras over all 5 Rubik's cube rotation experiments and plot the average metrics in Figure 5.17 (c). We observe that the *front left* and *back right* viewpoints result in lowest average pose error due to their closer proximity. The *top down* camera gives less reliable depth measurements and



(a) Example of a minimal robot cell **(b)** RGB-D views from cameras **(c)** Pose errors based on camera viewpoint

Figure 5.17: (a) As a proof-of-concept, we assembled a minimal robot cell for live demonstrations of our method with one RGB-D camera and the robot policy deployed at 2Hz. **(b)** The three different RGB-D camera viewpoints in our full robot cell used to collect FeelSight evaluation dataset. **(c)** Average pose error for known shape experiments based on camera viewpoint. We see that while the front and back cameras perform comparably, there are larger errors in the top-down camera as it is further away.

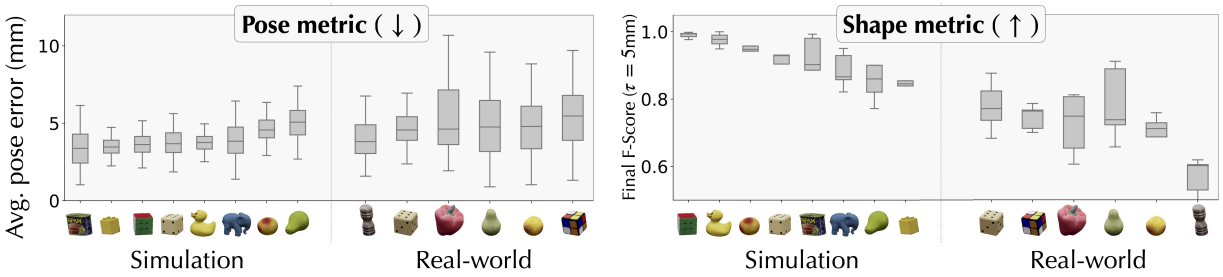


Figure 5.18: Pose **(left)** and shape **(right)** metrics for each object class, sorted in best-to-worst performance.

segmentation output, leading to almost 2x greater pose error.

Class-specific metrics. In Figure 5.18, we present our metrics for the SLAM results in Section 5.8.2, dividing based on object class. This helps us make some assessments on how object geometry and scale can affect our results. Some observations include:

- **Object symmetry.** Objects with symmetries about their rotation axis are challenging for our depth-based estimator. This leads to higher errors for the *peach* and *pear*, for example.
- **Object visibility.** Partial visibility of the large objects, such as the *pepper grinder*, affect the completeness of the reconstructions. Touch in this case is not advantageous since the finger gait does not span the length of the object to provide coverage.
- **Object scale.** Smaller-sized objects, such as the *peach*, may demonstrate better shape metrics as their scale is closer to the F-score threshold of 5 mm.

5.11 Additional visualizations

All experiments from the FeelSight dataset. In Figure 5.19 we illustrate all of the 70 visuo-tactile experiments that comprise our dataset. While both simulation and real data collection

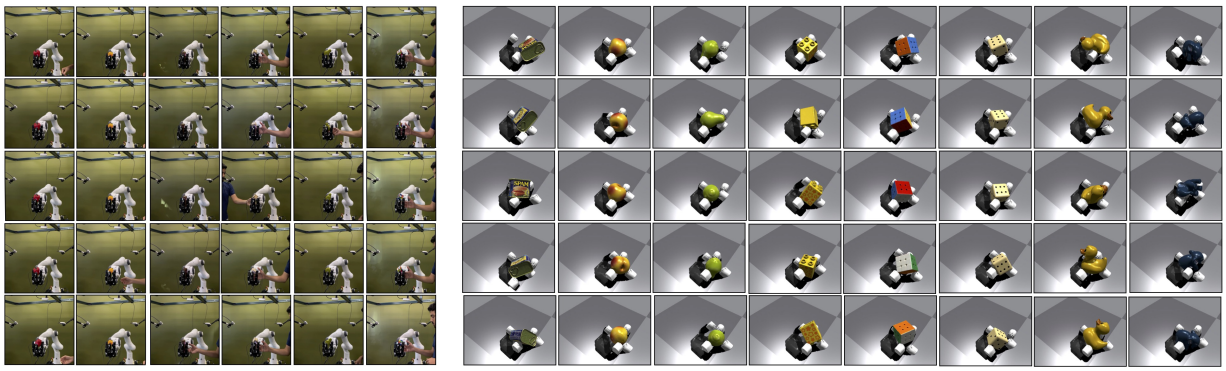
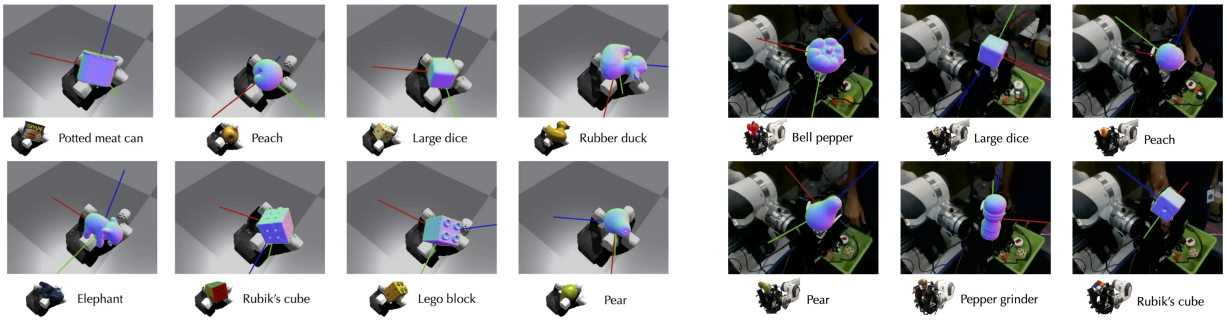


Figure 5.19: A collage depicting the entirety of the FeelSight dataset. We collect **(i)** 5 sequences each (row) in the real-world across 6 different objects (column), and **(ii)** 5 sequences each (row) in simulation across 8 different objects (column).

use the proprioception-driven policy [Qi+22], the policy generalizes better in simulation across the class of objects. Some objects in the real-world require a human-in-the-loop to assist with in-hand rotation; *e.g.* supporting cube-shaped objects from the bottom to occasionally prevent falling out of hand.

Additional neural tracking visualizations. Figure 5.20 shows rendering results from the experiments in Section 5.8.3 along with the pose axes. We see good alignment of the renderings when overlaid on the RGB camera frame.

Further visual segmentation results. Figure 5.21 shows additional qualitative results of visual segmentation for **(a)** real-world and **(b)** simulated rotations sequences.

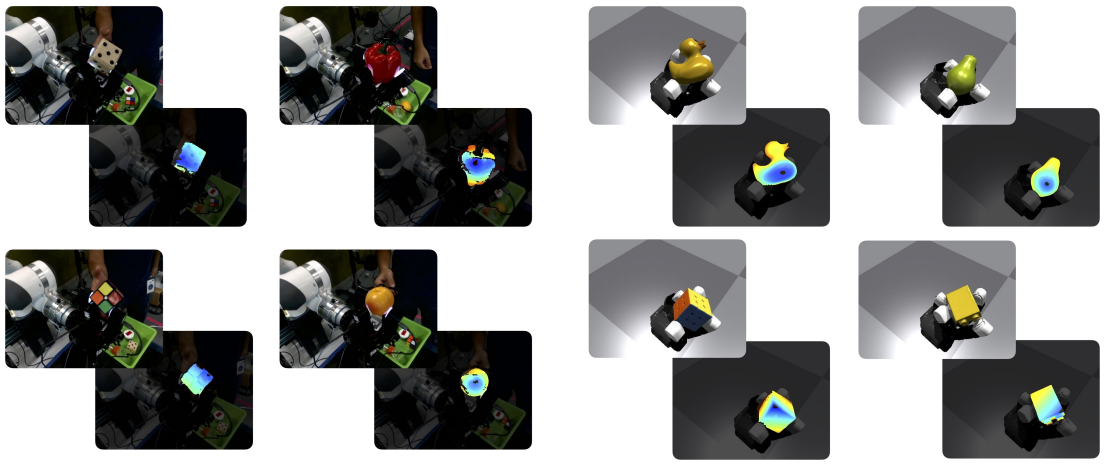


(a) Simulated pose tracking **(b)** Real-world pose tracking

Figure 5.20: Further visualizations of neural tracking experiments. These qualitatively complement the results from Section 5.8.3 for both **(a)** simulated and **(b)** real-world experiments.

5.11.1 Illustrating the role of touch

Sensor coverage visualized in SLAM. To illustrate the complementary nature of touch and vision, we color the reconstructed mesh regions based on their dominant sensing modality in



(a) Visual segmentation in the real-world **(b)** Visual segmentation in the simulation

Figure 5.21: Additional results on visual segmentation. Our segmentation module can accurately singulate the in-hand object in both **(a)** real-world and **(b)** simulated image sequences.

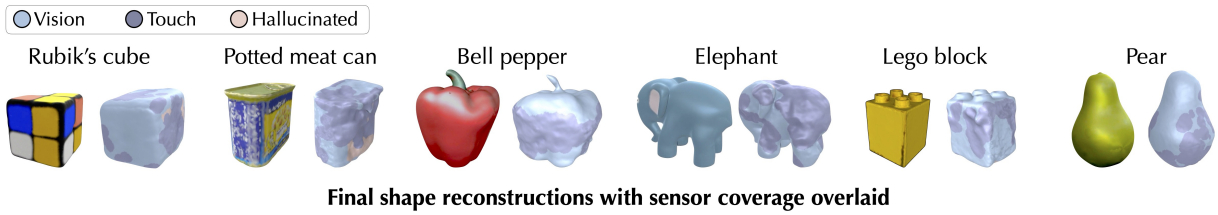
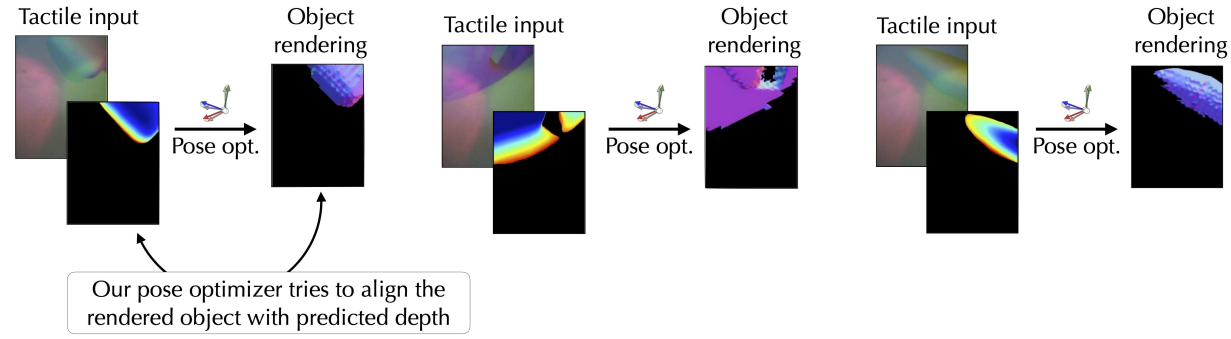


Figure 5.22: Sensor coverage illustrated in final mesh reconstructions of select objects—indicating vision, touch, and hallucinated regions.

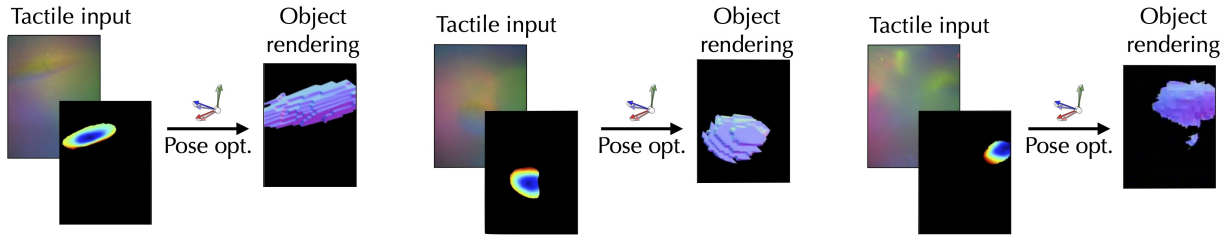
Figure 5.22. After running the SLAM experiments in Section 5.8.2, we first run marching-cubes on the final neural SDF. In the resultant mesh, we assign each vertices color based on if vision or touch is the nearest pointcloud measurement to it. In the case where there is no vision or touch pointcloud within a 5 mm radius, it is assigned as a *hallucinated* vertex. This is a demonstrable advantage of neural SDFs, where the network can extrapolate well based on information in the neighborhood of the query point. From the meshes in Figure 5.22 we see that while vision gets broad coverage of each object, there is considerable tactile signal from the interaction utilized for shape estimation.

Touch aligns local geometries with predicted depth. As described in Section 5.7.2, the pose optimizer *inverts* the neural field to back-propagate a loss in pose space [Yen+21; Suc+21; Zhu+22]. This has been illustrated in work such as iNeRF [Yen+21], where the rendered neural field attempts to match the image measurements via updates to the se(3) Lie algebra of the camera pose. As our framework leverages the idea that vision-based touch is just another perspective camera, we show how the rendered neural field matches with tactile depth features in Figure 5.23.

Each RGB image is first passed through the tactile transformer (Section 5.6.2) to output a pre-



(a) Tactile features in simulation



(b) Tactile features in the real-world

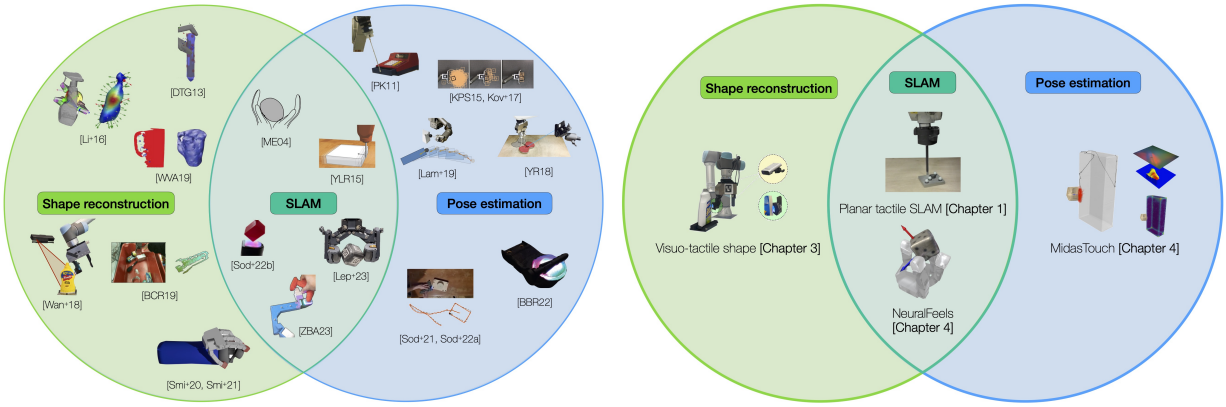
Figure 5.23: Six examples of tactile images compared against the neural field. We see that our tactile pose optimizer matches the predicted local geometry with the neural surface rendering. Thus, patches and edges predicted by touch appear in the rendering as well.

dicted tactile depthmap. Our pose optimizer aligns the neural rendering of the surface with the measured depthmap, based on 3D samples from the measured depthmap. Thus we can see that both in simulation (Figure 5.23 (a)) and the real-world (Figure 5.23 (b)), the edge and patch features predicted match well with the rendered object.

Chapter 6

Conclusion

Takeaways and future directions



(a) Prior work **(b)** Thesis contributions

Figure 6.1: Thesis contributions. **(a)** We build on prior and contemporaneous work in visuo-tactile perception, across planar pushing, sliding, non-prehensile manipulation, and dexterous interactions. **(b)** Our work spans across simpler 2D problems to complex dexterous interactions. *All images sourced from the respective publications.*

This thesis shows the potential of a multimodal system for manipulation that leverages pre-training and online learning. Spatial AI can further drive general dexterity on these robots, expanding what robots can do with their environments. The specific task of geometric perception is necessary for robots to understand what they are manipulating and how it responds to action. Information about geometry gives the planner and controller actionable feedback, and reveals the affordances of the scene.

An outcome of this thesis is high accuracy spatial understanding across different domains—planar pushing, sliding, and in-hand manipulation. This aligns well with the burgeoning interest across industry and academia in useful manipulation with humanoids, tabletop and mobile platforms. Indeed, SLAM for manipulation has been an active field of research for the past two

decades (Figure 6.1), and will be a key enabler for what robots can do in the future. In this Chapter, I highlight some key takeaways and end with some thoughts on future research directions for my work.

6.1 Key takeaways

A first takeaway is that *manipulation researchers should both maximize their sensory percept, and devise robust ways to process them.* In most tasks, occlusion is imminent and visual aliasing is expected. Without touch and kinematics we end up simply tolerating interaction rather than embracing it. Even with multimodal sensing, we require robust ways to process the 3D information. In this thesis, we demonstrate the application of SLAM and factor graphs towards vision, touch, kinematics and physics-based constraints.

Second, *tactile perception and simulation will play vital roles in robot-object interaction*. While the sensing and hardware may change over the years, the benefits of touch cannot be understated. This thesis, and notably Chapter 4, devise a tactile representation solely based on geometry. However, touch sensing gives us far more [Luo+17]—force, texture, hardness, friction, temperature, and high-frequency vibrations.

An equal partner is robot simulators, like IsaacGym [Mak+21] and MuJoCo [TET12]. When paired with tactile simulators like TACTO [Wan+22] and Taxim [SY22] we can scale touch data to a large corpus of interactions. In this thesis, we employ this technique to learn a tactile observation model (Chapter 4 and 5), but the benefits extend beyond that. The challenges going forward will be towards closing the sim-to-real gap, fine-tuning with real interactions, and standardizing touch sensing for broader adoption.

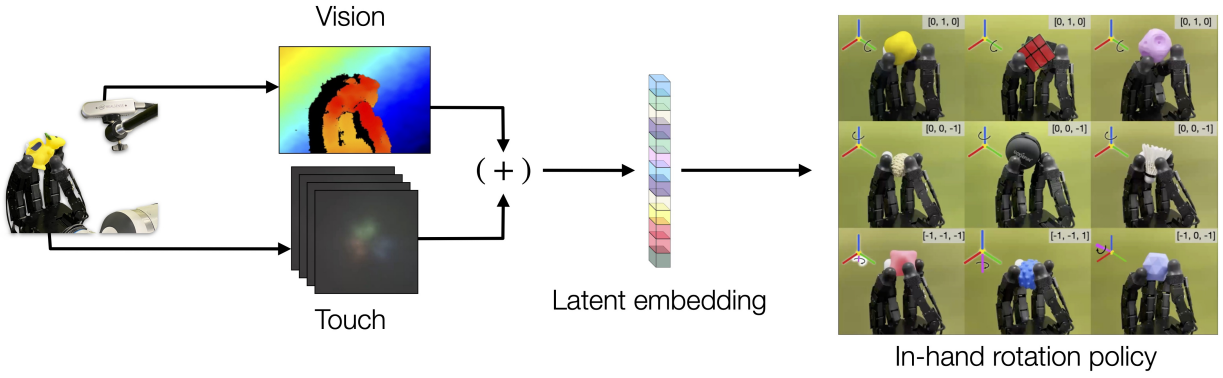


Figure 6.2: Visuo-tactile perception for RL. Fusing vision and touch as an embedding can greatly improve reinforcement learning policies. In the future, these can be further improved by grounding the policies with an online spatial representation. *All images sourced from the respective publications.*

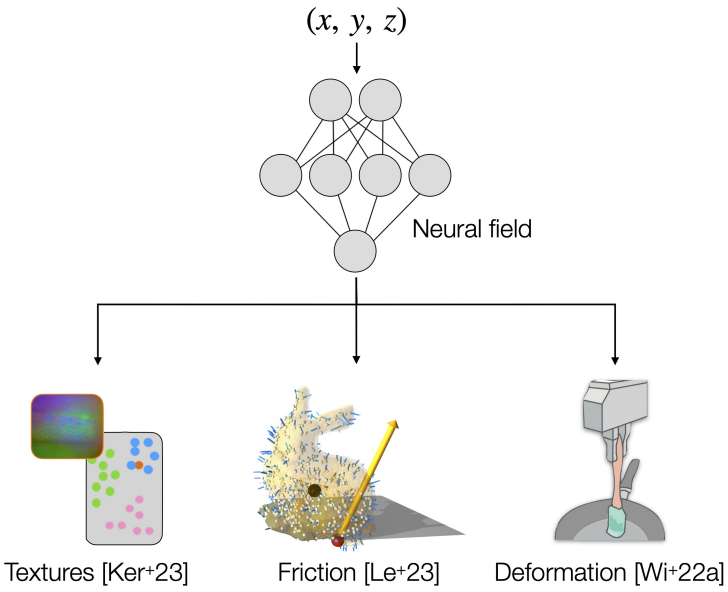


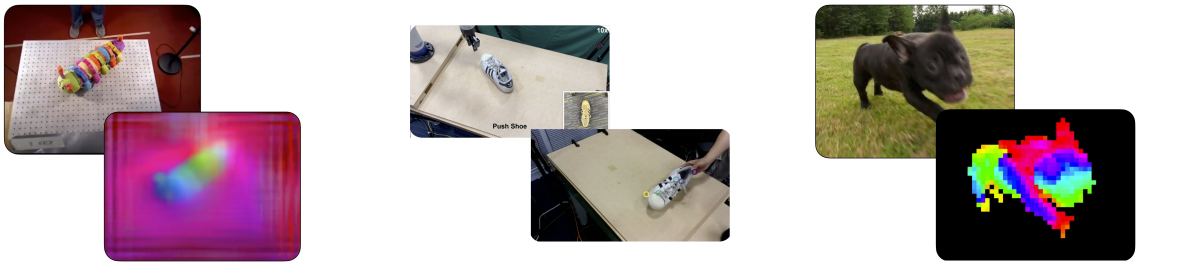
Figure 6.3: Latent state perception with neural fields. Future work can look towards expanding the state space of what we can infer, to include latent properties like textures [Ker+23], friction [Le +23], and object deformation [Wi+22a]. *All images sourced from the respective publications.*

6.2 Future directions

Perception-driven planning. NeuralFeels (Chapter 5) is relevant to manipulation researchers and practitioners who require spatial perception with a single camera and affordable tactile sensing. It can be extended to not just in-hand rotation, but many other object-centric manipulation tasks like in-hand reorientation [Che+23a], pick-and-place [Bau+23], insertion [Lep+23], nonprehensile sliding [Ker+22b], and planar pushing [Sur+21]. In the future, we hope to generalize to these different tasks and varied robot morphologies.

While not explored in this thesis, the direct benefit of an online SDF is the ability to seamlessly plan for dexterous interactions. Recent works demonstrate the benefit of apriori-known object point-clouds [Qi+23] and SDFs [Dri+22] for goal-conditioned planning, and running our perception stack in-the-loop is the next natural step. Specifically, Qi et al. [Qi+23] presents a system that *implicitly* fuses vision and touch towards learning a real-world rotation policy (Figure 6.2). Guzey et al. [Guz+23a] also optimize for both visual and tactile rewards for dexterity that enables challenging tasks. We can get the best of both worlds if we ground these policies with an online 3D representation.

Perceiving latent state. We consider geometry as just the starting point for neural models: interaction reveals latent properties like texture [Ker+22b], friction [Le +23], and object dynamics [SH21]. Neural fields can embed these latents as auxiliary optimization terms so as to benefit tasks that go beyond just geometry and pose (Figure 6.3). Applications can range from learning to manipulate inertially-significant objects (*e.g.* a hammer), to identifying a grasp point from



Multi-object learned descriptors [FMT18] **3D descriptor fields** [Wan+23] **Self-supervised foundation models** [Oqu+23]

Figure 6.4: Self-supervised learning for perception. Contrary to our approach of explicitly parameterizing object shape and pose, researchers have also explored implicit representations. Foundation models [Oqu+23] have the potential to be deployed out-of-the-box without fine-tuning. *All images sourced from the respective publications.*

local texture (*e.g.* a saucepan handle).

Implicit spatial representations.

Rather than explicitly representing object pose and shape, researchers have also looked into implicit representations derived from vision and touch feedback. These are typically spatial maps learned directly in pixel-space, via self-supervision from visual data [FMT18; Wan+23]. Foundation models [Oqu+23] trained on large datasets could possibly even be deployed out-of-the-box without fine-tuning.

Bibliography

This bibliography contains 208 references.

[ABL19] Kirsty Aquilina, David AW Barton, and Nathan F Lepora. "Shear-invariant sliding contact perception with a soft tactile sensor". In: *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 4283–4289 (page 40).

[Als+19] Alex Alspach, Kunimatsu Hashimoto, Naveen Kuppuswamy, and Russ Tedrake. "Soft-bubble: A highly compliant dense geometry tactile sensor for robot manipulation". In: *Proc. IEEE Intl. Conf. on Soft Robotics (RoboSoft)*. IEEE. 2019, pp. 597–604 (pages 2, 18, 30, 42, 65).

[Amb+21] Rares Ambrus, Vitor Guizilini, Naveen Kuppuswamy, Andrew Beaulieu, Adrien Gaidon, and Alex Alspach. "Monocular Depth Estimation for Soft Visuotactile Sensors". In: *Proc. IEEE Intl. Conf. on Soft Robotics (RoboSoft)*. 2021 (pages 18, 31, 48, 50).

[AMY21] Arpit Agarwal, Timothy Man, and Wenzhen Yuan. "Simulation of vision-based tactile sensors using physics based rendering". In: *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 1–7 (pages 2, 18, 30).

[Azi+22] Dejan Azinovi, Ricardo Martin-Brualla, Dan B Goldman, Matthias NieSSner, and Justus Thies. "Neural RGB-D surface reconstruction". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 6290–6301 (pages 48, 53).

[Bar+21] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. "Mip-nerf: A multiscale representation for anti-

aliasing neural radiance fields". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 5855–5864 (page 65).

[Bar+77] Harry G Barrow, Jay M Tenenbaum, Robert C Bolles, and Helen C Wolf. *Parametric correspondence and chamfer matching: Two new techniques for image matching*. Tech. rep. 1977, pp. 659–663 (page 24).

[Bau+20] Maria Bauza, Eric Valls, Bryan Lim, Theo Sechopoulos, and Alberto Rodriguez. "Tactile object pose estimation from the first touch with geometric contact rendering". In: *Proc. Conf. on Robot Learning, CoRL*. 2020 (pages 18, 30, 31).

[Bau+23] Maria Bauza, Antonia Bronars, Yifan Hou, Ian Taylor, Nikhil Chavan-Dafle, and Alberto Rodriguez. "simPLE: a visuotactile method learned in simulation to precisely pick, localize, regrasp, and place objects". In: *arXiv preprint arXiv:2307.13133* (2023) (pages 61, 64, 73).

[BBR22] Maria Bauza, Antonia Bronars, and Alberto Rodriguez. "Tac2Pose: Tactile Object Pose Estimation from the First Touch". In: *arXiv preprint arXiv:2204.11701* (2022) (pages 30, 31, 33, 34).

[BCR19] Maria Bauza, Oleguer Canal, and Alberto Rodriguez. "Tactile mapping and localization from high-resolution tactile imprints". In: *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 3811–3817 (pages 7, 17–19, 31, 48, 50, 56).

[BGD08] Alexander Bierbaum, Ilya Gubarev, and Rüdiger Dillmann. "Robust shape recovery for sparse contact location and normal data from haptic exploration". In: *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE. 2008, pp. 3200–3205 (page 19).

[Bim+16] Joao Bimbo, Shan Luo, Kaspar Althoefer, and Hongbin Liu. "In-hand object pose estimation using covariance-based tactile to geometry matching". In: *IEEE Robotics and Automation Letters (RA-L)* 1.1 (2016), pp. 570–577 (page 31).

[Bjö+13] Marten Björkman, Yasemin Bekiroglu, Virgile Högman, and Danica Kragic. "Enhancing visual perception of shape through tactile glances". In: *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE. 2013, pp. 3180–3186 (pages 18, 19).

[Bli82] James F Blinn. “A generalization of algebraic surface drawing”. In: *ACM Transactions on Graphics (TOG)* 1.3 (1982), pp. 235–256 (page 7).

[Bra+19] Samarth Brahmbhatt, Ankur Handa, James Hays, and Dieter Fox. “ContactGrasp: Functional multi-finger grasp synthesis from contact”. In: *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE. 2019, pp. 2386–2393 (pages 2, 29, 46, 57).

[Cad+16] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. “Past, present, and future of Simultaneous Localization and Mapping: Toward the robust-perception age”. In: *IEEE Trans. on Robotics (TRO)* 32.6 (2016), pp. 1309–1332 (pages 48, 65).

[Cal+17] Berk Calli, Arjun Singh, James Bruce, Aaron Walsman, Kurt Konolige, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. “Yale-CMU-Berkeley dataset for robotic manipulation research”. In: *Intl. J. of Robotics Research (IJRR)* 36.3 (2017), pp. 261–268 (pages 21, 24, 30, 33, 46, 51, 57).

[CGS19] Christopher Choy, JunYoung Gwak, and Silvio Savarese. “4D spatio-temporal ConvNets: Minkowski convolutional neural networks”. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 3075–3084 (pages 32, 33).

[Cha+22] Arkadeep Narayan Chaudhury, Timothy Man, Wenzhen Yuan, and Christopher G Atkeson. “Using Collocated Vision and Tactile Sensors for Visual Servoing and Localization”. In: *IEEE Robotics and Automation Letters (RA-L)* 7.2 (2022), pp. 3427–3434 (page 31).

[Che+23a] Tao Chen, Megha Tippur, Siyang Wu, Vikash Kumar, Edward Adelson, and Pulkit Agrawal. “Visual dexterity: In-hand dexterous manipulation from depth”. In: *ICML Workshop on New Frontiers in Learning, Control, and Dynamical Systems*. 2023 (pages 2, 42, 58, 61, 73).

[Che+23b] Yiting Chen, Ahmet Ercan Tekden, Marc Peter Deisenroth, and Yasemin Bekiroglu. “Sliding Touch-based Exploration for Modeling Unknown Object Shape with Multi-fingered Hands”. In: *arXiv preprint arXiv:2308.00576* (2023) (pages 3, 44).

[CHR18] Nikhil Chavan-Dafle, Rachel Holladay, and Alberto Rodriguez. “In-hand manipulation via motion cones”. In: *Proc. Robotics: Science and Systems (RSS)* (2018) (pages 2, 29).

[Cou10] Erwin Coumans. “Bullet physics engine”. In: *Open Source Software: http://bullet-physics. org* 1.3 (2010), p. 84 (page 13).

[CP10] Craig Corcoran and Robert Platt. “Tracking object pose and shape during robot manipulation based on tactile information”. In: *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*. Vol. 2. Citeseer. 2010 (page 31).

[CPK19] Christopher Choy, Jaesik Park, and Vladlen Koltun. “Fully convolutional geometric features”. In: *Proc. Intl. Conf. on Computer Vision (ICCV)*. 2019, pp. 8958–8966 (page 30).

[CRP13] Maxime Chalon, Jens Reinecke, and Martin Pfanne. “Online in-hand object localization”. In: *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE. 2013, pp. 2977–2984 (page 31).

[CXA22] Tao Chen, Jie Xu, and Pulkit Agrawal. “A system for general in-hand object re-orientation”. In: *Proc. Conf. on Robot Learning (CoRL)*. PMLR. 2022, pp. 297–307 (page 29).

[Dav18] Andrew J Davison. “FutureMapping: The computational structure of spatial AI systems”. In: *arXiv preprint arXiv:1803.11288* (2018) (page 42).

[DB06] Hugh Durrant-Whyte and Tim Bailey. “Simultaneous localization and mapping: part I”. In: *IEEE robotics & automation magazine* 13.2 (2006), pp. 99–110 (page 6).

[DD23] Eric Dexheimer and Andrew J Davison. “Learning a Depth Covariance Function”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 13122–13131 (page 66).

[Del+99] Frank Dellaert, Dieter Fox, Wolfram Burgard, and Sebastian Thrun. “Monte Carlo localization for mobile robots”. In: *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*. Vol. 2. IEEE. 1999, pp. 1322–1328 (pages 30, 31).

[Del12] Frank Dellaert. *Factor graphs and GTSAM: A hands-on introduction*. Tech. rep. Georgia Institute of Technology, 2012 (page 24).

[Den+21] Xinke Deng, Arsalan Mousavian, Yu Xiang, Fei Xia, Timothy Bretl, and Dieter Fox. “PoseRBPF: A Rao–Blackwellized particle filter for 6D object pose tracking”. In: *IEEE Trans. on Robotics (TRO)* 37.5 (2021), pp. 1328–1342 (pages 30, 34, 35, 40).

[DET17] Danny Driess, Peter Englert, and Marc Toussaint. “Active learning with query paths for tactile object shape exploration”. In: *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE. 2017, pp. 65–72 (page 19).

[DHT19] Danny Driess, Daniel Hennes, and Marc Toussaint. “Active Multi-Contact Continuous Tactile Exploration with Gaussian Process Differential Entropy”. In: *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 7844–7850 (page 7).

[DJ94] M-P Dubuisson and Anil K Jain. “A modified Hausdorff distance for object matching”. In: *Proc. Intl. Conf. on Pattern Recognition (ICPR)*. Vol. 1. IEEE. 1994, pp. 566–568 (page 13).

[DK17] Frank Dellaert and Michael Kaess. “Factor Graphs for Robot Perception”. In: *Foundations and Trends in Robotics* 6.1-2 (2017), pp. 1–139 (pages 11, 19, 22, 54, 55).

[DMR18] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. “Superpoint: Self-supervised interest point detection and description”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2018, pp. 224–236 (page 66).

[Don+18] 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: *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 1927–1934 (pages 2, 16, 18, 30, 31, 42).

[Dos+20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. “An image is worth 16x16 words: Transformers for image recognition at scale”. In: *arXiv preprint arXiv:2010.11929* (2020) (page 51).

[Dow+22] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. “Google Scanned Objects: A High-Quality Dataset of 3D Scanned Household Items”. In: *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)* (2022) (page 30).

[DR19] Siyuan Dong and Alberto Rodriguez. “Tactile-based insertion for dense box-packing”. In: *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE. 2019, pp. 7953–7960 (page 31).

[Dri+22] Danny Driess, Jung-Su Ha, Marc Toussaint, and Russ Tedrake. "Learning models as functionals of signed-distance fields for manipulation planning". In: *Conference on Robot Learning*. PMLR. 2022, pp. 245–255 (page 73).

[DS12] Mehmet R Dogar and Siddhartha S Srinivasa. "A planning framework for non-prehensile manipulation under clutter and uncertainty". In: *Autonomous Robots (AURO)* 33.3 (2012), pp. 217–236 (page 16).

[DTG11] Stanimir Dragiev, Marc Toussaint, and Michael Gienger. "Gaussian process implicit surfaces for shape estimation and grasping". In: *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE. 2011, pp. 2845–2850 (pages 7, 9, 19).

[DTG13] Stanimir Dragiev, Marc Toussaint, and Michael Gienger. "Uncertainty aware grasping and tactile exploration". In: *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE. 2013, pp. 113–119 (pages 7, 16).

[Dun+13] Kester Duncan, Sudeep Sarkar, Redwan Alqasemi, and Rajiv Dubey. "Multi-scale superquadric fitting for efficient shape and pose recovery of unknown objects". In: *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE. 2013, pp. 4238–4243 (page 6).

[EPF14] David Eigen, Christian Puhrsch, and Rob Fergus. "Depth map prediction from a single image using a multi-scale deep network". In: *Advances in Neural Information Processing Systems*. Vol. 3. 2014, pp. 2366–2374 (page 20).

[FMT18] Peter R Florence, Lucas Manuelli, and Russ Tedrake. "Dense object nets: Learning dense visual object descriptors by and for robotic manipulation". In: *arXiv preprint arXiv:1806.08756* (2018) (page 74).

[FRS13] Paul Furgale, Joern Rehder, and Roland Siegwart. "Unified temporal and spatial calibration for multi-sensor systems". In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2013, pp. 1280–1286 (page 45).

[Gan+20] Gabriela Zarzar Gandler, Carl Henrik Ek, Mårten Björkman, Rustam Stolkin, and Yasemin Bekiroglu. "Object shape estimation and modeling, based on sparse Gaussian process implicit surfaces, combining visual data and tactile exploration". In: *J. of Robotics and Autonomous Systems (RAS)* 126 (2020), p. 103433 (page 18).

[Gao+22] Ruohan Gao, Zilin Si, Yen-Yu Chang, Samuel Clarke, Jeannette Bohg, Li Fei-Fei, Wenzhen Yuan, and Jiajun Wu. "ObjectFolder 2.0: A Multisensory Object Dataset

for Sim2Real Transfer". In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2022 (page 31).

[Gar+14] Sergio Garrido-Jurado, Rafael Muñoz-Salinas, Francisco José Madrid-Cuevas, and Manuel Jesús Marín-Jiménez. "Automatic generation and detection of highly reliable fiducial markers under occlusion". In: *Pattern Recognition* 47.6 (2014), pp. 2280–2292 (page 45).

[Gei+13] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. "Vision meets robotics: The KITTI dataset". In: *The International Journal of Robotics Research* 32.11 (2013), pp. 1231–1237 (page 46).

[Gro+23] Phillip Grote, Joaquim Ortiz-Haro, Marc Toussaint, and Ozgur S Oguz. "Neural Field Representations of Articulated Objects for Robotic Manipulation Planning". In: *arXiv preprint arXiv:2309.07620* (2023) (page 42).

[GRP91] Suresh Goyal, Andy Ruina, and Jim Papadopoulos. "Planar sliding with dry friction part 1. limit surface and moment function". In: *Wear* 143.2 (1991), pp. 307–330 (pages 6, 8, 11).

[Guz+23a] Irmak Guzey, Yinlong Dai, Ben Evans, Soumith Chintala, and Lerrel Pinto. "See to touch: Learning tactile dexterity through visual incentives". In: *arXiv preprint arXiv:2309.12300* (2023) (pages 2, 73).

[Guz+23b] Irmak Guzey, Ben Evans, Soumith Chintala, and Lerrel Pinto. "Dexterity from Touch: Self-Supervised Pre-Training of Tactile Representations with Robotic Play". In: *arXiv preprint arXiv:2303.12076* (2023) (pages 2, 42).

[Han+14] Ankur Handa, Thomas Whelan, John McDonald, and Andrew J Davison. "A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM". In: *2014 IEEE international conference on Robotics and automation (ICRA)*. IEEE. 2014, pp. 1524–1531 (page 63).

[Han+22] Ankur Handa et al. "DeXtreme: Transfer of Agile In-Hand Manipulation from Simulation to Reality". In: *arXiv* (2022) (pages 2, 42, 58, 61).

[Haw+19] Jeff Hawkins, Marcus Lewis, Mirko Klukas, Scott Purdy, and Subutai Ahmad. "A framework for intelligence and cortical function based on grid cells in the neocortex". In: *Frontiers in neural circuits* 12 (2019), p. 121 (page 30).

[HBM23] Carolina Higuera, Byron Boots, and Mustafa Mukadam. “Learning to Read Braille: Bridging the Tactile Reality Gap with Diffusion Models”. In: *arXiv preprint arXiv:2304.01182* (2023) (pages 3, 65).

[HE07] Hannah B Helbig and Marc O Ernst. “Optimal integration of shape information from vision and touch”. In: *Experimental brain research* 179.4 (2007), pp. 595–606 (pages 1, 18, 42).

[Heb+11] Paul Hebert, Nicolas Hudson, Jeremy Ma, and Joel Burdick. “Fusion of stereo vision, force-torque, and joint sensors for estimation of in-hand object location”. In: *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE. 2011, pp. 5935–5941 (page 7).

[Hel61] Joseph Heller. *Catch-22: a novel*. Vol. 4. Simon and Schuster, 1961 (page 29).

[HK19] Ming Hsiao and Michael Kaess. “MH-iSAM2: Multi-hypothesis iSAM using Bayes Tree and Hypo-tree”. In: *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 1274–1280 (page 16).

[Hod+18] Tomas Hodan, Frank Michel, Eric Brachmann, Wadim Kehl, Anders GlentBuch, Dirk Kraft, Bertram Drost, Joel Vidal, Stephan Ihrke, Xenophon Zabulis, et al. “BOP: Benchmark for 6D object pose estimation”. In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 19–34 (page 46).

[Hon+23] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. “LRM: Large Reconstruction Model for Single Image to 3D”. In: *arXiv preprint arXiv:2311.04400* (2023) (page 65).

[HS05] Aaron Hertzmann and Steven M Seitz. “Example-based photometric stereo: Shape reconstruction with general, varying BRDFs”. In: *IEEE Trans. Pattern Anal. Machine Intell.* 27.8 (2005), pp. 1254–1264 (page 18).

[IBK14] Jarmo Ilonen, Jeannette Bohg, and Ville Kyrki. “Three-dimensional object reconstruction of symmetric objects by fusing visual and tactile sensing”. In: *Intl. J. of Robotics Research (IJRR)* 33.2 (2014), pp. 321–341 (page 18).

[Iza+17] Gregory Izatt, Geronimo Mirano, Edward Adelson, and Russ Tedrake. “Tracking objects with point clouds from vision and touch”. In: *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 4000–4007 (page 7).

[JA09] Micah K. Johnson and Edward H. Adelson. “Retrographic sensing for the measurement of surface texture and shape”. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2009, pp. 1070–1077 (pages 18, 31).

[Jam+16] Nawid Jamali, Carlo Ciliberto, Lorenzo Rosasco, and Lorenzo Natale. “Active perception: Building objects’ models using tactile exploration”. In: *Proc. IEEE-RAS Intl. Conf. on Humanoid Robots (Humanoids)*. IEEE. 2016, pp. 179–185 (page 19).

[Jam+20] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. “RLBench: The robot learning benchmark and learning environment”. In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 3019–3026 (page 46).

[Joh+11] Micah K. Johnson, Forrester Cole, Alvin Raj, and Edward H. Adelson. “Microgeometry Capture Using an Elastomeric Sensor”. In: *ACM Trans. Graph.* 30.4 (July 2011) (pages 18, 31).

[JRB18] Rico Jonschkowski, Divyam Rastogi, and Oliver Brock. “Differentiable particle filters: End-to-end learning with algorithmic priors”. In: *Proc. Robotics: Science and Systems (RSS)*. 2018 (page 40).

[Kae+12] Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, John Leonard, and Frank Dellaert. “iSAM2: Incremental smoothing and mapping using the Bayes tree”. In: *Intl. J. of Robotics Research (IJRR)* 31.2 (2012), pp. 216–235 (pages 7, 11, 13, 24).

[KB14] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014) (page 53).

[Ker+22a] Justin Kerr, Letian Fu, Huang Huang, Yahav Avigal, Matthew Tancik, Jeffrey Ichnowski, Angjoo Kanazawa, and Ken Goldberg. “Evo-NeRF: Evolving NeRF for sequential robot grasping of transparent objects”. In: *6th Annual Conference on Robot Learning*. 2022 (page 42).

[Ker+22b] Justin Kerr, Huang Huang, Albert Wilcox, Ryan Hoque, Jeffrey Ichnowski, Roberto Calandra, and Ken Goldberg. “Learning Self-Supervised Representations from Vision and Touch for Active Sliding Perception of Deformable Surfaces”. In: *arXiv preprint arXiv:2209.13042* (2022) (pages 29, 73).

[Ker+23] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. "3D Gaussian splatting for real-time radiance field rendering". In: *ACM Transactions on Graphics (ToG)* 42.4 (2023), pp. 1–14 (pages 65, 73).

[Kes+23] Leonid Keselman, Katherine Shih, Martial Hebert, and Aaron Steinfeld. "Optimizing Algorithms From Pairwise User Preferences". In: *arXiv preprint arXiv:2308.04571* (2023) (page 62).

[Kir+23] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. "Segment anything". In: *arXiv preprint arXiv:2304.02643* (2023) (pages 48–50).

[KK13] Soohwan Kim and Jonghyuk Kim. "Continuous occupancy maps using overlapping local gaussian processes". In: *2013 IEEE/RSJ international conference on intelligent robots and systems*. IEEE. 2013, pp. 4709–4714 (pages 7, 10).

[KLM85] Roberta L Klatzky, Susan J Lederman, and Victoria A Metzger. "Identifying objects by touch: An expert system". In: *Perception & Psychophysics* 37.4 (1985), pp. 299–302 (pages 1, 5).

[Kna+17] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. "Tanks and temples: Benchmarking large-scale scene reconstruction". In: *ACM Transactions on Graphics (ToG)* 36.4 (2017), pp. 1–13 (page 56).

[Kom21] Jacek Komorowski. "MinkLoc3D: Point cloud based large-scale place recognition". In: *Proc. Winter Conf. on Applications of Computer Vision (WACV)*. 2021, pp. 1790–1799 (pages 30, 32–34).

[Kov+17] Michael C Koval, Matthew Klingensmith, Siddhartha S Srinivasa, Nancy S Pollard, and Michael Kaess. "The manifold particle filter for state estimation on high-dimensional implicit manifolds". In: *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 4673–4680 (page 31).

[KPP22] Tarik Kelestemur, Robert Platt, and Taskin Padir. "Tactile Pose Estimation and Policy Learning for Unknown Object Manipulation". In: *arXiv preprint arXiv:2203.10685* (2022) (page 31).

[KPS15] Michael C Koval, Nancy S Pollard, and Siddhartha S Srinivasa. "Pose estimation for planar contact manipulation with manifold particle filters". In: *Intl. J. of Robotics Research (IJRR)* 34.7 (2015), pp. 922–945 (page 7).

[KR22] Sangwoon Kim and Alberto Rodriguez. “Active Extrinsic Contact Sensing: Application to General Peg-in-Hole Insertion”. In: *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE. 2022 (page 31).

[Kup+19] Naveen Kuppuswamy, Alejandro Castro, Calder Phillips-Grafflin, Alex Alspach, and Russ Tedrake. “Fast model-based contact patch and pose estimation for highly deformable dense-geometry tactile sensors”. In: *IEEE Robotics and Automation Letters (RA-L)* 5.2 (2019), pp. 1811–1818 (page 31).

[Lab+22] Yann Labbé, Lucas Manuelli, Arsalan Mousavian, Stephen Tyree, Stan Birchfield, Jonathan Tremblay, Justin Carpentier, Mathieu Aubry, Dieter Fox, and Josef Sivic. “Megapose: 6d pose estimation of novel objects via render & compare”. In: *arXiv preprint arXiv:2212.06870* (2022) (page 61).

[Lai+16] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. “Deeper depth prediction with fully convolutional residual networks”. In: *Proc. Intl. Conf. on 3D Vision (3DV)*. IEEE. 2016, pp. 239–248 (pages 20, 33).

[Lam+19] Alexander Sasha Lambert, Mustafa Mukadam, Balakumar Sundaralingam, Nathan Ratliff, Byron Boots, and Dieter Fox. “Joint inference of kinematic and force trajectories with visuo-tactile sensing”. In: *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 3165–3171 (pages 3, 7, 8, 12, 44, 61).

[Lam+20] Mike Lambeta, Po-Wei Chou, Stephen Tian, Brian Yang, Benjamin Maloon, Victoria Rose Most, Dave Stroud, Raymond Santos, Ahmad Byagowi, Gregg Kammerer, et al. “DIGIT: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation”. In: *IEEE Robotics and Automation Letters (RA-L)* 5.3 (2020), pp. 3838–3845 (pages 2, 18, 30, 31, 42–45, 51).

[LC87] William E Lorensen and Harvey E Cline. “Marching cubes: A high resolution 3D surface construction algorithm”. In: *ACM SIGGRAPH Computer Graphics* 21.4 (1987), pp. 163–169 (page 24).

[LC91] Soo Hong Lee and MR Cutkosky. “Fixture planning with friction”. In: *J. of Manufacturing Science and Engineering* 113.3 (1991) (pages 8, 11).

[Le +23] Simon Le Cleac’h, Hong-Xing Yu, Michelle Guo, Taylor Howell, Ruohan Gao, Jiajun Wu, Zachary Manchester, and Mac Schwager. “Differentiable physics simulation of dynamics-augmented neural objects”. In: *IEEE Robotics and Automation Letters* 8.5 (2023), pp. 2780–2787 (pages 42, 73).

[Lee+19] Bhoram Lee, Clark Zhang, Zonghao Huang, and Daniel D Lee. "Online continuous mapping using Gaussian process implicit surfaces". In: *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 6884–6890 (pages 7, 10, 19, 23, 24).

[Lep+23] Marion Lepert, Chaoyi Pan, Shenli Yuan, Rika Antonova, and Jeannette Bohg. "In-Hand Manipulation of Unknown Objects with Tactile Sensing for Insertion". In: *Embracing Contacts-Workshop at ICRA 2023*. 2023 (pages 3, 43, 73).

[Li+14] Rui Li, Robert Platt, Wenzhen Yuan, Andreas ten Pas, Nathan Roscup, Mandayam A Srinivasan, and Edward Adelson. "Localization and manipulation of small parts using GelSight tactile sensing". In: *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE. 2014, pp. 3988–3993 (pages 30, 31).

[Li+16] Miao Li, Kaiyu Hang, Danica Kragic, and Aude Billard. "Dexterous grasping under shape uncertainty". In: *J. of Robotics and Autonomous Systems (RAS)* 75 (2016), pp. 352–364 (page 7).

[Li+23] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. "Neuralangelo: High-Fidelity Neural Surface Reconstruction". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 8456–8465 (page 42).

[Lin+17] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. "Feature pyramid networks for object detection". In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 2117–2125 (page 33).

[Lin+21] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. "BARF: Bundle-adjusting neural radiance fields". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 5741–5751 (page 47).

[LK87] Susan J Lederman and Roberta L Klatzky. "Hand movements: A window into haptic object recognition". In: *Cognitive psychology* 19.3 (1987), pp. 342–368 (pages 1, 30).

[LL86] Jack M Loomis and Susan J Lederman. "Tactual perception". In: *Handbook of perception and human performances* 2.2 (1986), p. 2 (page 1).

[LMT92] Kevin M Lynch, Hitoshi Maekawa, and Kazuo Tanie. “Manipulation and active sensing by pushing using tactile feedback.” In: *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. Vol. 1. 1992 (pages 6, 11).

[Luo+15] Shan Luo, Wenxuan Mou, Kaspar Althoefer, and Hongbin Liu. “Localizing the object contact through matching tactile features with visual map”. In: *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE. 2015, pp. 3903–3908 (page 31).

[Luo+17] Shan Luo, Joao Bimbo, Ravinder Dahiya, and Hongbin Liu. “Robotic tactile perception of object properties: A review”. In: *Mechatronics* 48 (2017), pp. 54–67 (pages 2, 6, 72).

[Mak+21] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. “Isaac Gym: High performance GPU-based physics simulation for robot learning”. In: *arXiv preprint arXiv:2108.10470* (2021) (pages 45, 46, 57, 72).

[Mar+07] F Landis Markley, Yang Cheng, John L Crassidis, and Yaakov Oshman. “Averaging quaternions”. In: *Journal of Guidance, Control, and Dynamics* 30.4 (2007), pp. 1193–1197 (page 36).

[Mar+13] Uriel Martinez-Hernandez, Giorgio Metta, Tony J Dodd, Tony J Prescott, Lorenzo Natale, and Nathan F Lepora. “Active contour following to explore object shape with robot touch”. In: *2013 World Haptics Conference (WHC)*. IEEE. 2013, pp. 341–346 (page 7).

[Mas86] Matthew T Mason. “Mechanics and planning of manipulator pushing operations”. In: *Intl. J. of Robotics Research (IJRR)* 5.3 (1986), pp. 53–71 (page 15).

[Mat19] Matthew Matl. *Pyrender*. https://github.com/mmatl/pyrender. 2019 (page 25).

[ME04] Mark Moll and Michael A Erdmann. “Reconstructing the shape and motion of unknown objects with active tactile sensors”. In: *Algorithmic Foundations of Robotics V*. Springer, 2004, pp. 293–309 (pages 3, 6, 43).

[Mei+11] Martin Meier, Matthias Schopfer, Robert Haschke, and Helge Ritter. “A probabilistic approach to tactile shape reconstruction”. In: *IEEE Trans. on Robotics (TRO)* 27.3 (2011), pp. 630–635 (pages 6, 7).

[Mil+21] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. “NeRF: Representing scenes as neural radiance fields

for view synthesis". In: *Communications of the ACM* 65.1 (2021), pp. 99–106 (pages 42, 48, 53).

[Mor88] Hans Moravec. *Mind children: The future of robot and human intelligence*. Harvard University Press, 1988 (pages 1, 41).

[Muk+18] Mustafa Mukadam, Jing Dong, Xinyan Yan, Frank Dellaert, and Byron Boots. "Continuous-time Gaussian process motion planning via probabilistic inference". In: *Intl. J. of Robotics Research (IJRR)* 37.11 (2018), pp. 1319–1340 (page 19).

[Mül+22] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. "Instant neural graphics primitives with a multiresolution hash encoding". In: *ACM Transactions on Graphics (ToG)* 41.4 (2022), pp. 1–15 (pages 42, 47, 48, 65).

[Ope+18] OpenAI et al. "Learning Dexterous In-Hand Manipulation". In: *CoRR* (2018). URL: http://arxiv.org/abs/1808.00177 (pages 2, 29, 32, 42).

[Ope+19] OpenAI et al. "Solving Rubik's Cube with a robot hand". In: *arXiv preprint arXiv:1910.07113* (2019) (pages 2, 42).

[Oqu+23] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. "Dinov2: Learning robust visual features without supervision". In: *arXiv preprint arXiv:2304.07193* (2023) (pages xvii, 74).

[Ort+22] Joseph Ortiz, Alexander Clegg, Jing Dong, Edgar Sucar, David Novotny, Michael Zollhoefer, and Mustafa Mukadam. "iSDF: Real-Time Neural Signed Distance Fields for Robot Perception". In: *arXiv preprint arXiv:2204.02296* (2022) (pages xiii, 2, 42, 43, 48, 52, 53).

[Ott+16] Simon Ottenhaus, Martin Miller, David Schiebener, Nikolaus Vahrenkamp, and Tamim Asfour. "Local implicit surface estimation for haptic exploration". In: *Proc. IEEE-RAS Intl. Conf. on Humanoid Robots (Humanoids)*. IEEE. 2016, pp. 850–856 (page 19).

[Pad+20] Akhil Padmanabha, Frederik Ebert, Stephen Tian, Roberto Calandra, Chelsea Finn, and Sergey Levine. "OmniTact: A multi-directional high-resolution touch sensor". In: *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 618–624 (pages 2, 18, 30, 42).

[Par+19] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. "DeepSDF: Learning continuous signed distance functions for shape representation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 165–174 (pages 42, 65).

[Pin+22] Luis Pineda et al. "Theseus: A Library for Differentiable Nonlinear Optimization". In: *Advances in Neural Information Processing Systems* (2022) (pages 34, 47, 55, 65).

[PK11] Anna Petrovskaya and Oussama Khatib. "Global localization of objects via touch". In: *IEEE Trans. on Robotics (TRO)* 27.3 (2011), pp. 569–585 (pages 7, 31).

[PRH11] Zachary Pezzementi, Caitlin Reyda, and Gregory D Hager. "Object mapping, recognition, and localization from tactile geometry". In: *2011 IEEE International Conference on Robotics and Automation*. IEEE. 2011, pp. 5942–5948 (page 31).

[Qi+17] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. "PointNet: Deep learning on point sets for 3D classification and segmentation". In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 652–660 (page 32).

[Qi+22] Haozhi Qi, Ashish Kumar, Roberto Calandra, Yi Ma, and Jitendra Malik. "In-hand object rotation via rapid motor adaptation". In: *Conference on Robot Learning*. PMLR. 2022, pp. 1722–1732 (pages 2, 42, 43, 46, 55, 58, 68).

[Qi+23] Haozhi Qi, Brent Yi, Sudharshan Suresh, Mike Lambeta, Yi Ma, Roberto Calandra, and Jitendra Malik. "General In-Hand Object Rotation with Vision and Touch". In: *Conference on Robot Learning*. PMLR. 2023, pp. 1722–1732 (pages 2, 42, 46, 58, 65, 73).

[Ras03] Carl Edward Rasmussen. "Gaussian processes in machine learning". In: *Summer School on Machine Learning*. Springer. 2003, pp. 63–71 (pages 7, 9, 10, 21, 22).

[RBK21] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. "Vision transformers for dense prediction". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 12179–12188 (pages 50, 51).

[Rev23] Revopoint. *Revopoint POP 3 3D Scanner*. 2023. URL: https://www.revopoint3d.com/pop3-3d-scanner/ (pages 46, 57).

[RHL14] David M Rosen, Guoquan Huang, and John J Leonard. "Inference over heterogeneous finite-/infinite-dimensional systems using factor graphs and Gaussian pro-

cesses". In: *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE. 2014, pp. 1261–1268 (page 19).

[RLC22] Antoni Rosinol, John J Leonard, and Luca Carlone. "NeRF-SLAM: Real-time dense monocular SLAM with neural radiance fields". In: *arXiv preprint arXiv:2210.13641* (2022) (pages 47, 48).

[RYH10] Ananth Ranganathan, Ming-Hsuan Yang, and Jeffrey Ho. "Online sparse Gaussian process regression and its applications". In: *IEEE Trans. on Image Processing* 20.2 (2010), pp. 391–404 (page 19).

[SB90] Franc Solina and Ruzena Bajcsy. "Recovery of parametric models from range images: The case for superquadrics with global deformations". In: *IEEE Trans. Pattern Anal. Machine Intell.* 12.2 (1990), pp. 131–147 (page 6).

[Sch+17a] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. "DBSCAN revisited, revisited: why and how you should (still) use DBSCAN". In: *ACM Transactions on Database Systems (TODS)* 42.3 (2017), pp. 1–21 (page 36).

[Sch+17b] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. "Proximal policy optimization algorithms". In: *arXiv preprint arXiv:1707.06347* (2017) (page 46).

[SCS17] Brad Saund, Shiyuan Chen, and Reid Simmons. "Touch based localization of parts for high precision manufacturing". In: *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 378–385 (page 31).

[SF16] Johannes L Schonberger and Jan-Michael Frahm. "Structure-from-Motion revisited". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 4104–4113 (page 48).

[SG06] Edward Snelson and Zoubin Ghahramani. "Sparse Gaussian processes using pseudo-inputs". In: *Advances in Neural Information Processing Systems* 18 (2006), pp. 1259–1266 (page 7).

[SH18] Balakumar Sundaralingam and Tucker Hermans. "Geometric in-hand regrasp planning: Alternating optimization of finger gaits and in-grasp manipulation". In: *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 231–238 (pages 2, 29).

[SH21] Balakumar Sundaralingam and Tucker Hermans. “In-hand object-dynamics inference using tactile fingertips”. In: *IEEE Transactions on Robotics* 37.4 (2021), pp. 1115–1126 (page 73).

[She+21] Yu She, Shaoxiong Wang, Siyuan Dong, Neha Sunil, Alberto Rodriguez, and Edward Adelson. “Cable manipulation with a tactile-reactive gripper”. In: *Intl. J. of Robotics Research (IJRR)* 40.12-14 (2021), pp. 1385–1401 (pages 2, 29, 42).

[Shi+17] Jian Shi, J Zachary Woodruff, Paul B Umbanhowar, and Kevin M Lynch. “Dynamic in-hand sliding manipulation”. In: *IEEE Trans. on Robotics (TRO)* 33.4 (2017), pp. 778–795 (pages 2, 29).

[Smi+20] Edward J Smith, Roberto Calandra, Adriana Romero, Georgia Gkioxari, David Meger, Jitendra Malik, and Michal Drozdzal. “3D shape reconstruction from vision and touch”. In: *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*. 2020 (pages 3, 17–19, 44, 64).

[Smi+21] Edward J Smith, David Meger, Luis Pineda, Roberto Calandra, Jitendra Malik, Adriana Romero, and Michal Drozdzal. “Active 3D Shape Reconstruction from Vision and Touch”. In: *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*. 2021 (pages 17, 18, 64).

[SNF15] Tanner Schmidt, Richard Newcombe, and Dieter Fox. “DART: dense articulated real-time tracking with consumer depth cameras”. In: *Autonomous Robots (AURO)* 39.3 (2015), pp. 239–258 (page 6).

[Sod+21] Paloma Sodhi, Michael Kaess, Mustafa Mukadam, and Stuart Anderson. “Learning tactile models for factor graph-based estimation”. In: *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 13686–13692 (pages 3, 7, 30, 31, 36, 44, 61).

[Sod+22a] Paloma Sodhi, Eric Dexheimer, Mustafa Mukadam, Stuart Anderson, and Michael Kaess. “LEO: Learning energy-based models in factor graph optimization”. In: *Conference on Robot Learning*. PMLR. 2022, pp. 234–244 (page 7).

[Sod+22b] Paloma Sodhi, Michael Kaess, Mustafa Mukadam, and Stuart Anderson. “PatchGraph: In-hand tactile tracking with learned surface normals”. In: *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2022 (pages 2, 18, 30, 31, 36, 40, 42, 48, 50).

[SS20] Johannes A Stork and Todor Stoyanov. “Ensemble of sparse Gaussian process experts for implicit surface mapping with streaming data”. In: *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 10758–10764 (pages 7, 10, 13, 19, 23, 24).

[Str+14] Claudius Strub, Florentin Wörgötter, Helge Ritter, and Yulia Sandamirskaya. “Correcting pose estimates during tactile exploration of object shape: a neuro-robotic study”. In: *4th International Conference on Development and Learning and on Epigenetic Robotics*. IEEE. 2014, pp. 26–33 (pages 3, 7, 43).

[Suc+21] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J Davison. “iMAP: Implicit mapping and positioning in real-time”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 6229–6238 (pages xiii, 2, 42, 43, 47, 53, 55, 64, 69).

[Sur+21] Sudharshan Suresh, Maria Bauza, Kuan-Ting Yu, Joshua G Mangelson, Alberto Rodriguez, and Michael Kaess. “Tactile SLAM: Real-time inference of shape and pose from planar pushing”. In: *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*. May 2021 (pages 3, 19, 27, 40, 43, 64, 65, 73).

[Sur+22a] Sudharshan Suresh, Zilin Si, Stuart Anderson, Michael Kaess, and Mustafa Mukadam. “MidasTouch: Monte-Carlo inference over distributions across sliding touch”. In: *6th Annual Conference on Robot Learning*. 2022 (pages 4, 42, 51, 61).

[Sur+22b] Sudharshan Suresh, Zilin Si, Joshua G Mangelson, Wenzhen Yuan, and Michael Kaess. “ShapeMap 3-D: Efficient shape mapping through dense touch and vision”. In: *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*. Philadelphia, PA, USA, May 2022 (pages 4, 31, 33, 40, 42, 44, 48, 50, 51, 64).

[Sur+23] Sudharshan Suresh et al. “Neural feels with neural fields: Visuo-tactile perception for in-hand manipulation”. In: *arXiv preprint arXiv:2312.1346*. Dec. 2023 (page 4).

[SY22] Zilin Si and Wenzhen Yuan. “Taxim: An Example-based Simulation Model for GelSight Tactile Sensors”. In: *IEEE Robotics and Automation Letters (RA-L)* (2022) (pages 2, 18, 20, 25, 30, 72).

[Tan+20] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. “Fourier features let networks learn high frequency functions in low dimensional

domains". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 7537–7547 (page 48).

[Tat+19] Maxim Tatarchenko, Stephan R Richter, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. "What do single-view 3D reconstruction networks learn?" In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 3405–3414 (page 56).

[TET12] Emanuel Todorov, Tom Erez, and Yuval Tassa. "Mujoco: A physics engine for model-based control". In: *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE. 2012, pp. 5026–5033 (page 72).

[Thr02a] Sebastian Thrun. "Particle Filters in Robotics". In: *UAI*. Vol. 2. Citeseer. 2002, pp. 511–518 (page 31).

[Thr02b] Sebastian Thrun. "Probabilistic robotics". In: *Communications of the ACM* 45.3 (2002), pp. 52–57 (page 36).

[Tre+18] Jonathan Tremblay, Thang To, Balakumar Sundaralingam, Yu Xiang, Dieter Fox, and Stan Birchfield. "Deep object pose estimation for semantic robotic grasping of household objects". In: *arXiv preprint arXiv:1809.10790* (2018) (page 56).

[Tre+23] Jonathan Tremblay, Bowen Wen, Valts Blukis, Balakumar Sundaralingam, Stephen Tyree, and Stan Birchfield. "Diff-DOPE: Differentiable Deep Object Pose Estimation". In: *arXiv preprint arXiv:2310.00463* (2023) (page 56).

[UL18] Mikaela Angelina Uy and Gim Hee Lee. "PointNetVLAD: Deep point cloud based retrieval for large-scale place recognition". In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 4470–4479 (pages 30, 32, 33).

[Var+17] Jacob Varley, Chad DeChant, Adam Richardson, Joaquín Ruales, and Peter Allen. "Shape completion enabled robotic grasping". In: *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE. 2017, pp. 2442–2447 (pages 6, 18, 27).

[VH08] Laurens Van der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE." In: *Journal of machine learning research* 9.11 (2008) (pages xii, 34).

[Wan+18] 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: *Proc. IEEE/RSJ Intl. Conf. on Intelligent*

Robots and Systems (IROS). IEEE. 2018, pp. 1606–1613 (pages 3, 16–19, 27, 44, 65).

[Wan+21] Shaoxiong Wang, Yu She, Branden Romero, and Edward Adelson. "GelSight Wedge: Measuring High-Resolution 3D Contact Geometry with a Compact Robot Finger". In: *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE. 2021 (pages 2, 18, 30, 31, 42, 48, 50, 65).

[Wan+22] Shaoxiong Wang, Mike Maroje Lambeta, Po-Wei Chou, and Roberto Calandra. "TACTO: A Fast, Flexible, and Open-source Simulator for High-resolution Vision-based Tactile Sensors". In: *IEEE Robotics and Automation Letters (RA-L)* (2022) (pages 2, 18, 30, 31, 33, 42, 45, 47, 51, 53, 72).

[Wan+23] Yixuan Wang, Zhuoran Li, Mingtong Zhang, Katherine Driggs-Campbell, Jiajun Wu, Li Fei-Fei, and Yunzhu Li. "D3 Fields: Dynamic 3D Descriptor Fields for Zero-Shot Generalizable Robotic Manipulation". In: *arXiv preprint arXiv:2309.16118* (2023) (page 74).

[War+18] Benjamin Ward-Cherrier, Nicholas Pestell, Luke Cramphorn, Benjamin Winstone, Maria Elena Giannaccini, Jonathan Rossiter, and Nathan F Lepora. "The TacTip family: Soft optical tactile sensors with 3D-printed biomimetic morphologies". In: *Soft robotics* 5.2 (2018), pp. 216–227 (pages 2, 18, 30, 42).

[Wen+23] Bowen Wen, Jonathan Tremblay, Valts Blukis, Stephen Tyree, Thomas Müller, Alex Evans, Dieter Fox, Jan Kautz, and Stan Birchfield. "BundleSDF: Neural 6-DoF Tracking and 3D Reconstruction of Unknown Objects". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 606–617 (pages 42, 64).

[WF07] Oliver Williams and Andrew Fitzgibbon. "Gaussian process implicit surfaces". In: *Gaussian Proc. in Practice* (2007), pp. 1–4 (pages 6, 7, 9, 19, 22).

[Wi+22a] Youngsun Wi, Pete Florence, Andy Zeng, and Nima Fazeli. "VIRDO: Visio-tactile implicit representations of deformable objects". In: *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE. 2022 (pages 40, 73).

[Wi+22b] Youngsun Wi, Andy Zeng, Pete Florence, and Nima Fazeli. "VIRDO++: Real-World, Visuo-tactile Dynamics and Perception of Deformable Objects". In: *arXiv preprint arXiv:2210.03701* (2022) (page 42).

[Won23] Wonik Robotics. *Allegro Hand*. 2023. URL: http://wiki.wonikrobotics.com/AllegroHandWiki/index.php/Allegro_Hand (pages 43, 45).

[WSE19] Jinkun Wang, Tixiao Shan, and Brendan Englot. "Underwater terrain reconstruction from forward-looking sonar imagery". In: *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 3471–3477 (page 19).

[Wu+23] Chao-Yuan Wu, Justin Johnson, Jitendra Malik, Christoph Feichtenhofer, and Georgia Gkioxari. "Multiview Compressive Coding for 3D Reconstruction". In: *arXiv:2301.08247* (2023) (page 65).

[Xia+17] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. "PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes". In: *arXiv preprint arXiv:1711.00199* (2017) (page 56).

[Xie+22] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. "Neural fields in visual computing and beyond". In: *Computer Graphics Forum*. Vol. 41. Wiley Online Library. 2022, pp. 641–676 (page 42).

[XPF18] XPFly1989. *FCRN: Fully Convolutional Residual Network for Depth Estimation*. https://github.com/XPFly1989/FCRN. 2018 (page 20).

[YA16] Akihiko Yamaguchi and Christopher G Atkeson. "Combining finger vision and optical tactile sensing: Reducing and handling errors while cutting vegetables". In: *Proc. IEEE-RAS Intl. Conf. on Humanoid Robots (Humanoids)*. IEEE. 2016, pp. 1045–1051 (pages 18, 30).

[YDA17] Wenzhen Yuan, Siyuan Dong, and Edward H Adelson. "GelSight: High-resolution robot tactile sensors for estimating geometry and force". In: *Sensors* 17.12 (2017), p. 2762 (pages 2, 16, 18, 21, 30, 31, 42, 50, 65).

[Yen+21] Lin Yen-Chen, Pete Florence, Jonathan T Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. "iNeRF: Inverting neural radiance fields for pose estimation". In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2021, pp. 1323–1330 (pages 42, 47, 55, 69).

[Yi+16] Zhengkun Yi, Roberto Calandra, Filipe Veiga, Herke van Hoof, Tucker Hermans, Yilei Zhang, and Jan Peters. "Active tactile object exploration with Gaussian pro-

cesses". In: *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE. 2016, pp. 4925–4930 (pages 7, 19).

[YIB17] Xinyan Yan, Vadim Indelman, and Byron Boots. "Incremental sparse GP regression for continuous-time trajectory estimation and mapping". In: *J. of Robotics and Autonomous Systems (RAS)* 87 (2017), pp. 120–132 (page 19).

[Yin+23] Zhao-Heng Yin, Binghao Huang, Yuzhe Qin, Qifeng Chen, and Xiaolong Wang. "Rotating without Seeing: Towards In-hand Dexterity through Touch". In: *arXiv preprint arXiv:2303.10880* (2023) (pages 2, 42, 46).

[YLR15] Kuan-Ting Yu, John Leonard, and Alberto Rodriguez. "Shape and pose recovery from planar pushing". In: *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE. 2015, pp. 1208–1215 (pages x, 3, 6, 8, 12, 13, 16, 43).

[YR18] Kuan–Ting Yu and Alberto Rodriguez. "Realtime state estimation with tactile and visual sensing: application to planar manipulation". In: *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 7778–7785 (pages 3, 7, 8, 11, 31, 44, 61).

[Yu+16] Kuan-Ting Yu, Maria Bauza, Nima Fazeli, and Alberto Rodriguez. "More than a million ways to be pushed: a high-fidelity experimental dataset of planar pushing". In: *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE. 2016, pp. 30–37 (pages ix, 9, 10, 13).

[Yu+21] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. "PixelNeRF: Neural radiance fields from one or few images". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 4578–4587 (page 42).

[Yua+15] Wenzhen Yuan, Rui Li, Mandayam A Srinivasan, and Edward H Adelson. "Measurement of shear and slip with a GelSight tactile sensor". In: *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE. 2015, pp. 304–311 (page 36).

[ZBA23] Jialiang Zhao, Maria Bauza, and Edward H Adelson. "FingerSLAM: Closed-loop Unknown Object Localization and Reconstruction from Visuo-tactile Feedback". In: *arXiv preprint arXiv:2303.07997* (2023) (pages 44, 58, 64, 65).

[Zha+23] Chaoning Zhang, Dongshen Han, Yu Qiao, Jung Uk Kim, Sung-Ho Bae, Seungkyu Lee, and Choong Seon Hong. "Faster Segment Anything: Towards Lightweight SAM for Mobile Applications". In: *arXiv preprint arXiv:2306.14289* (2023) (page 50).

[Zho+22] Shaohong Zhong, Alessandro Albini, Oiwi Parker Jones, Perla Maiolino, and Ingmar Posner. "Touching a NeRF: Leveraging Neural Radiance Fields for Tactile Sensory Data Generation". In: *6th Annual Conference on Robot Learning*. 2022 (page 42).

[Zhu+22] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R Oswald, and Marc Pollefeys. "NICE-SLAM: Neural implicit scalable encoding for SLAM". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 12786–12796 (pages xiii, 2, 42, 43, 47, 55, 64, 69).

[ZLT13] Li Zhang, Siwei Lyu, and Jeff Trinkle. "A dynamic Bayesian approach to real-time estimation and filtering in grasp acquisition". In: *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE. 2013, pp. 85–92 (page 7).

[ZT12] Li Zhang and Jeffrey C Trinkle. "The application of particle filtering to grasping acquisition with visual occlusion and tactile sensing". In: *2012 IEEE International Conference on Robotics and Automation*. IEEE. 2012, pp. 3805–3812 (page 31).