MidasTouch: Monte-Carlo inference over distributions across sliding touch

Anonymous Author(s)
Affiliation
Address
email

Abstract: We present MidasTouch, a tactile perception system for online global localization of a vision-based touch sensor sliding on an object surface. This framework takes in posed tactile images over time, and outputs an evolving distribution of sensor pose on the object's surface, without the need for visual priors. Our key insight is to estimate local surface geometry with tactile sensing, learn a compact representation for it, and disambiguate these signals over a long time horizon. The backbone of MidasTouch is a Monte-Carlo particle filter, with a measurement model based on a tactile code network learned from tactile simulation. This network, inspired by LIDAR place recognition, compactly summarizes local surface geometries. These generated codes are efficiently compared against a pre-computed tactile codebook per-object, to update the pose distribution. We further release the YCB-Slide dataset of real-world and simulated forceful sliding interactions between a vision-based tactile sensor and standard YCB objects. While single-touch localization can be inherently ambiguous, we can quickly localize our sensor by traversing salient surface geometries.

Keywords: Tactile perception, Localization, 3D deep learning

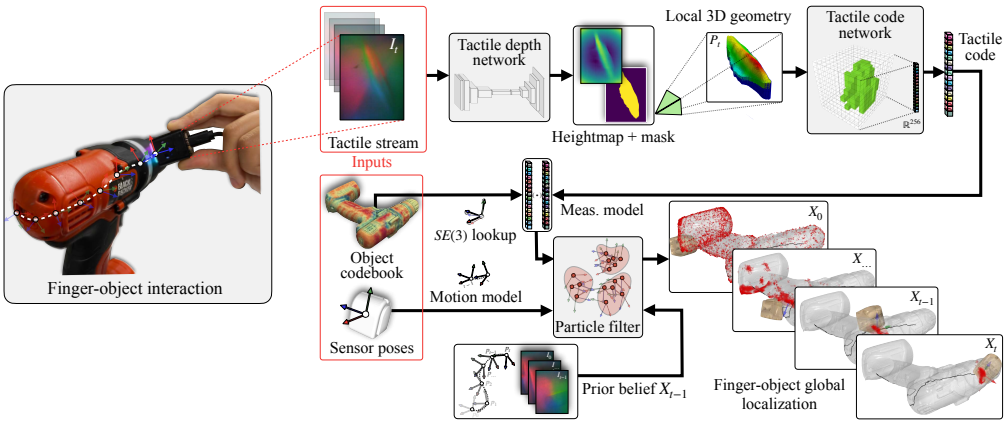

Figure 1: MidasTouch performs online global localization of a vision-based touch sensor on an object surface during sliding interactions. Given posed tactile images over time, this system leverages local surface geometry within a nonparametric particle filter to generate an evolving distribution of sensor pose on the object's surface.

1 Introduction

Interactive perception is the Catch-22 [1] 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 [2–4], in-hand manoeuvres [5, 6], finger-gaiting [7], and multi-finger enclosure [8]. With high-dimensional vision-based tactile sensors [9–11], 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.

Submitted to the 6th Conference on Robot Learning (CoRL 2022). Do not distribute.

The form-factor of vision-based tactile sensors has restricted prior methods to small parts [12–14] or local tracking [15, 16]. For everyday objects, we can leverage compositionality [17]; 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 [18].

We approach this as an analogue to mobile robot Monte-Carlo filtering [19], 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 [20, 21]. While priors from vision can bootstrap our system [22], 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 [11] interactions across standard YCB objects [20] 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 DIGIT and YCB object interactions for evaluation and benchmarking.

Our framework relies on recent developments in tactile sensing, rendering, and robotics: **(i)** vision-based tactile sensors [9–11, 23–27] like the GelSight and DIGIT have the requisite spatial acuity to discern local geometric features, **(ii)** tactile simulation [28–30] 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 [31–33] spawned by ubiquitous LIDAR and RGB-D data that can be extended to tactile sensing.

2 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 [34–39]. 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 [19, 40]. With tactile arrays, local patch measurements are integrated for planar estimation [41, 42] or for 3D alignment [43]. The synergy of vision and touch gives much needed global context [44, 45] but is outside our current work's scope.

With vision-based touch, Li et al. [12] 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 [46], or auto-encoder [11, 15] networks. Additionally, local tracking has been explored with online factor-graph optimization [15, 16, 47]. The aforementioned approaches are unimodal and rely on good pose initializations; MidasTouch is nonparametric and requires no such knowledge. Additionally, Kelestemur et al. [48] show category-level object pose estimation with a parallel jaw gripper.

Closely related is Tac2Pose [13, 14], which performs pose estimation of small-parts with the GelSlim [10]. It produces a distribution of object poses learned in tactile simulation from contact shapes. Similarly, Gao et al. [49] 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 interme-

diate is a learned embedding from either RGB [11, 15] or binary contact masks [13, 14]. Realistic tactile simulators, like TACTO [28], enable training these models to the scale and generality of arbitrary real-world interactions. For example, Bauza et al. [14] 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 [16, 50]. For vision-based touch, 3D geometry is obtained either via photometric stereo [9, 51, 52], or image-to-heightmap models [16, 27, 53–55]. 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 [32, 33]. Following the seminal PointNet [56], Choy et al. [57] 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 [33]. We show tactile geometries can be compacted into codes with the same architecture, for easy frame-to-model queries.

3 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 $\mathbf{x}_t \in SE(3)$ in the object-centric frame. Such pose context is useful for downstream planning and control, for instance manipulating an object in-hand [5]. 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, $\mathbf{z}_t = \left\{\mathbf{I}_t \in \mathbb{R}^{240\times320\times3},\ \mathbf{p}_t \in SE(3)\right\}$. 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 $\mathbf{x}_t$, but later show the benefit of visual priors.

4 Global localization during sliding touch

MidasTouch comprises three distinct modules, as illustrated in Figure 1: a tactile depth network (TDN: Section 4.1), tactile code network (TCN: Section 4.2), and particle filter (Section 4.3). At a high-level, the TDN first converts a tactile image $\mathbf{I}_t$ into its local 3D geometry $\mathbf{P}_t$ via a learned observation model. The 3D information is then condensed into a tactile code $\mathbf{E}_t$ 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`. Additional implementation details can be found in the Appendix.

4.1 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 [55, 58], and train it in a supervised fashion 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 [28], we render a large collection of DIGIT images with ground-truth heightmaps. The images are from simulated interaction with 40 YCB [20] 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

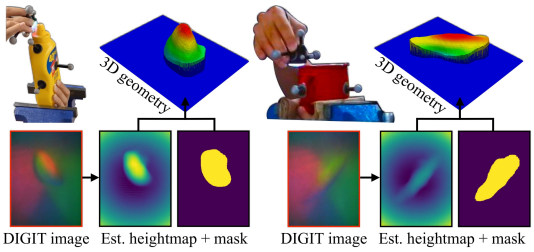

Figure 2: Real-world DIGIT images from interactions in the `YCB-Slide` dataset (more in Appendix). Our tactile depth network (Section 4.1) is trained in simulation, and predicts 3D geometry given an input tactile image.

calibrate TACTO with real-world data from multiple independent DIGITs, and apply random lighting augmentations (refer Appendix). 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 $\mathbf{I}_t$ through the network to get heightmap $\mathbf{H}_t$. To remove non-contact areas we compute a mask $\mathbf{C}_t$ by depth thresholding, and get $\hat{\mathbf{H}}_t = \mathbf{H}_t \odot \mathbf{C}_t$. Finally, the heightmap is reprojected to 3D via the camera's known perspective projection model $\hat{\mathbf{H}}_t \mapsto \mathbf{P}_t$. Evaluations on the test set show heightmap RMSE of 0.135 mm with respect to ground-truth. Figure 2 shows examples of the TDN on real-world DIGIT interactions.

4.2 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. [14]. 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 [32] and MinkLoc3D [33], 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 [57], performs state-of-the-art point cloud retrieval through sparse 3D convolutions.

Network and training: The architecture comprises of three components: (i) Voxelization of $\mathbf{P}_t$ into a quantized sparse tensor $\hat{\mathbf{P}}_t = \{\langle \hat{x}_i, \hat{y}_i, \hat{z}_i, 0/1 \rangle\}$, (ii) Feature pyramid network [59] for per-voxel local features $\hat{\mathbf{P}}_t^f = \{\langle \hat{x}_i, \hat{y}_i, \hat{z}_i, \hat{\mathbf{f}}_i \in \mathbb{R}^{256} \rangle\}$, and (iii) Generalized-mean pooling for point cloud embedding vector $\mathbf{E}_t \in \mathbb{R}^{256}$. We use the pre-trained weights, learned from four comprehensive LIDAR datasets [33], 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.

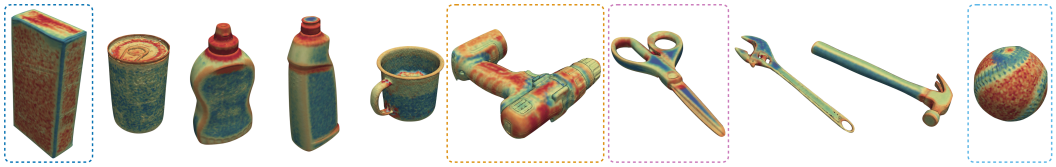

Figure 3: Tactile codebook per object visualized as a spectral colorspace map using t-SNE [60]. Each codebook comprises of 50k densely sampled poses with their corresponding 256-dimensional tactile code. Similar hues denote sensor poses that elicit similar tactile codes. We can clearly delineate local geometric features: edges (`sugar_box`) , ridges (`power_drill`), corners (`scissors`), and complex texture (`baseball`).

Tactile codebook: Once we have a code $\mathbf{E}_t$, we would like to efficiently compare this with a dense set of contacts. Inspired by prior work [14, 22], 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 (refer Appendix). 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: $\mathcal{C}_o = \{\langle \mathbf{p}^{[m]}, \mathbf{E}^{[m]} \rangle\}_{m=1}^{M}$, where $\mathbf{p}^{[m]}$ are $SE(3)$ sensor poses in codebook and $\mathbf{E}^{[m]}$ are corresponding tactile codes. Figure 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.

For nearest-neighbor lookup, we build a KD-Tree with the 6-element pose vectors: $\{[\mathbf{p}_{\text{trans}}^{[m]}, \ \alpha \log(\mathbf{p}_{\text{rot}}^{[m]})]\}_{m=1}^{M}$, where $\log(\cdot)$ is the $SO(3)$ Logarithm map and $\alpha = 0.01$ is an empirical scaling factor. Thus, given a candidate pose $\mathbf{x}_t^{[i]}$ we do not have to render and obtain its

corresponding code, but instead just perform a pose-space lookup $\mathcal{C}_o(\mathbf{x}_t^{[i]})$. 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 Figure 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. We compare against embeddings from tactile images in the Appendix.

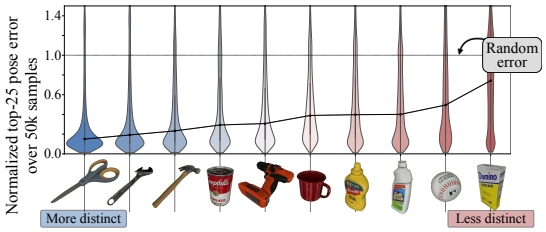

Figure 4: Pose-error for 50k single-touch queries on YCB objects, in ascending order (normalized with respect to random touch). For each query, we get the top-25 highest scores from the tactile codebook $\mathcal{C}_o$, and compute their minimum pose-error with respect to ground-truth. We observe tools with salient geometries to be easier to localize versus objects that exhibit symmetry.

4.3 Filtering over posed touch

The particle filter approximates the posterior distribution of the sensor pose as a set $\mathbf{X}_t = \{\langle \mathbf{x}_t^{[i]}, w_t^{[i]} \rangle\}_{i=1}^{N_t}$, where $\mathbf{x}_t^{[i]}$ 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 $\mathbf{X}_0$ is sampled from a coarse prior, which can (optionally) be obtained from vision [22]. Estimating contact location from vision is inherently noisy and can be ambiguous due to object symmetries. In our experiments, 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 $\mathbf{x}_0^{\text{gt}}$ as $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{x}_0^{\text{gt}}, \begin{bmatrix} \sigma_{\text{trans}} & 0 \\ 0 & \sigma_{\text{rot}} \end{bmatrix})$, 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.

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 $\mathbf{X}_{t-1}$ forward by sampling from a state transition probability:

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

Additionally, odometry can also be from marker flow [61] or image-to-image tracking [15, 16].

Measurement update: In this step, we update the particle weights based on how their tactile codes match the current measurement $\mathbf{E}_t$, as a function of cosine distance. At each timestep, we lookup the codebook for particle codes (Section 4.2) and perform a matrix-vector multiplication:

$$w_t^{[i]} = p(\mathbf{z}_t \mid \mathbf{x}_t^{[i]}) \sim \text{softmax}\left(\frac{\mathbf{E}_t \cdot \mathcal{C}_o(\mathbf{x}_t^{[i]})}{||\mathbf{E}_t|| \cdot ||\mathcal{C}_o(\mathbf{x}_t^{[i]})||}\right) \tag{2}$$

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 $\mathbf{X}_t$ from the proposal distribution with a probability proportional to their weights. Low-variance resampling [62] 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 DB-SCAN [63]. We average the cluster positions and quaternions [64] to get a hypothesis set $h_t : \{\mathbf{x}_t^1 \dots \mathbf{x}_t^H\}$.

5 The `YCB-Slide` dataset

To evaluate MidasTouch, and enable further research in tactile sensing, we introduce our `YCB-Slide` dataset (more in Appendix). It comprises of DIGIT sliding interactions on the 10 YCB objects from Section 4. 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, and ground-truth mesh models, and ground-truth heightmaps + contact masks (simulation only).

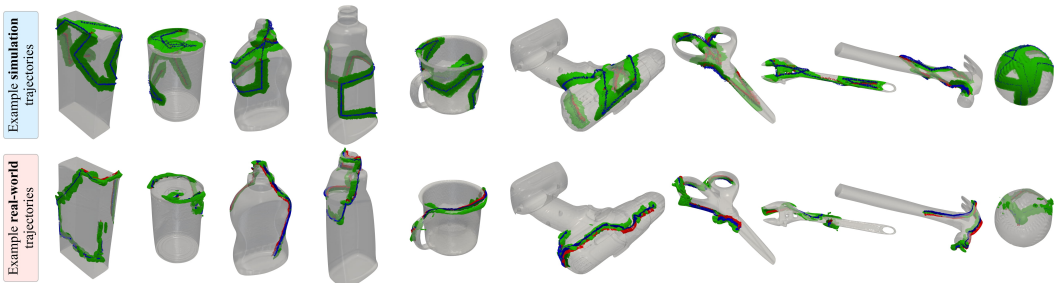

Figure 5: Example sliding trajectories from simulated and real trials on the 10 YCB objects. Overlaid in green are the local 3D geometries captured by the tactile sensor, and the contact poses as RGB coordinate axes.

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 $\sigma_{\text{trans}} = 0.5$mm, $\sigma_{\text{rot}} = 1°$. 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 Figure 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 5 and 6.

Figure 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.

6 Experimental results

Simulation: We evaluate MidasTouch over the 50 trajectories collected in Section 5. As the filter is non-deterministic, each trajectory is run 10 times for averaged results over 500 trials. Figure 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 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.

Figure 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 Figure 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 at-

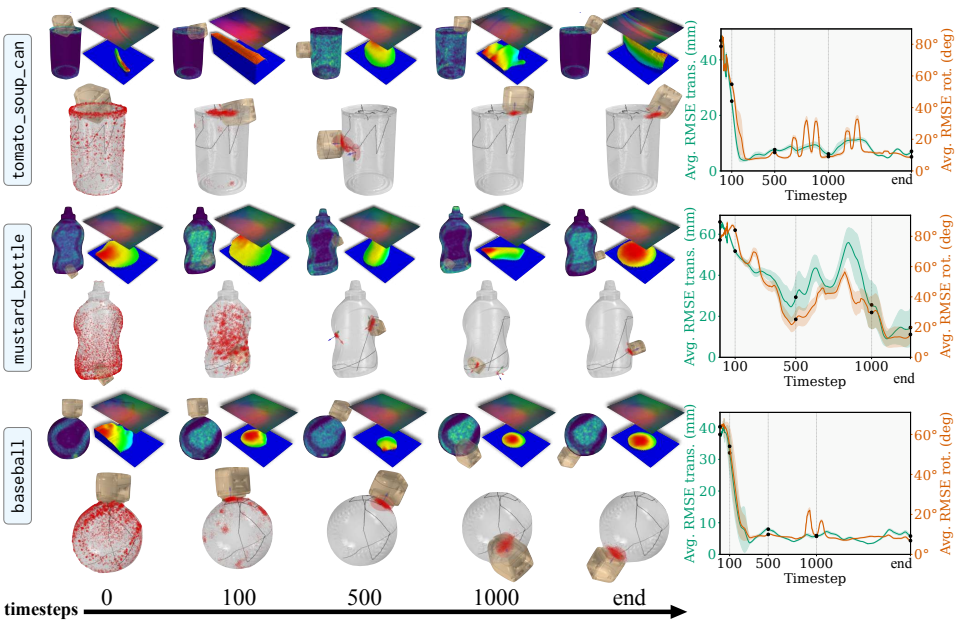

Figure 7: Snapshots of simulated sliding results on three YCB objects. For each row: **[top]** the tactile images, local geometries, and heatmap of pose likelihood with respect to the tactile codebook, **[bottom]** pose distribution evolving over time, and converging to the most-likely hypothesis after encountering salient geometries, **[right]** average translation and rotation RMSE of the distribution over time with variance over 10 trials.

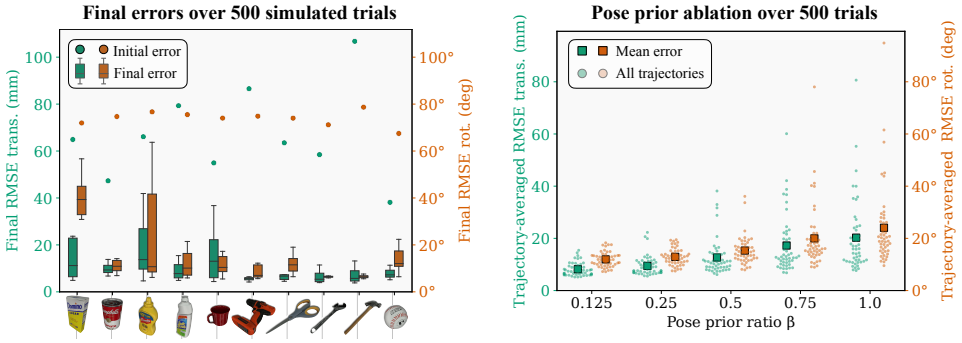

Figure 8: [left] Boxplot of final trajectory error over 500 simulation trials. For each object, we plot the averaged initial and final RMSE for the particle set. We observe better convergence for tools with salient geometries, as opposed to symmetric objects. **[right]** Ablation over initial pose uncertainty, to show lower average trajectory error with better visual priors. With a weaker initialization ($\beta = 1.0$), outliers in pose-error are more prevalent.

tributed 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 (Figure 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 5, with a tighter pose prior of $\beta = 0.5$. In Figure 10 we present three representative trajectories that show the hypotheses converge to the ground-truth pose. The accumulated statistics in Figure 9 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

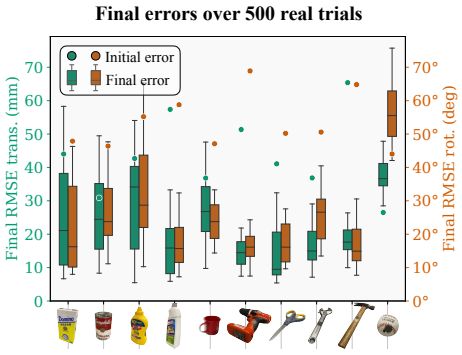

Figure 9: Boxplot of final error over 500 real-world trials from the 50 `YCB-Slide` trajectories.

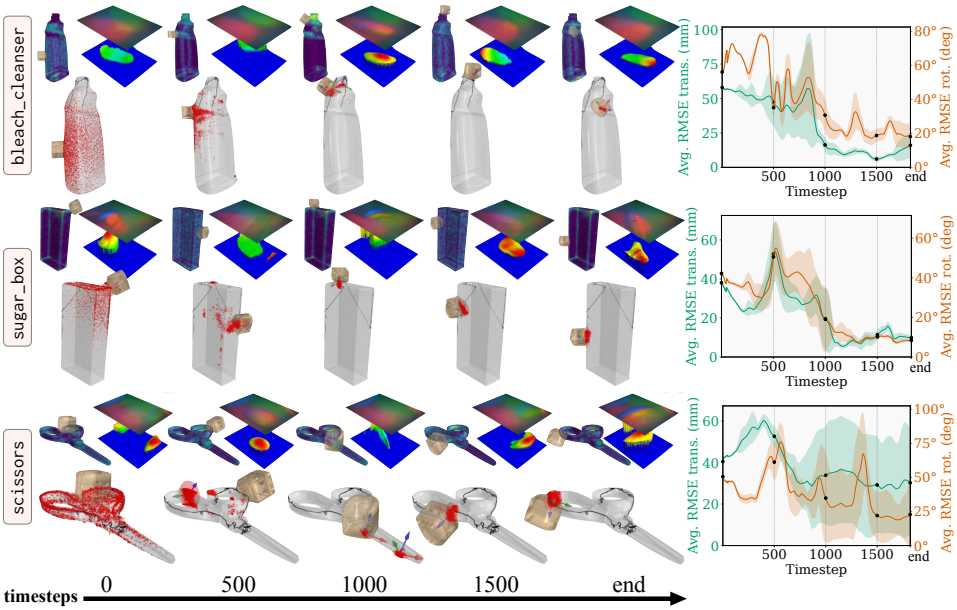

Figure 10: Snapshots of real-world sliding results on three YCB objects. For each row: **[top]** the tactile images, local geometries, and heatmap of pose likelihood with respect to the tactile codebook, **[bottom]** pose distribution evolving over time, and converging to the most-likely hypothesis after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

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 detected from real tactile images, effectively meaning we are trying to localize on a featureless sphere. These failure modes are highlighted in Figure 11.

7 Conclusion and limitations

In this work, 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.

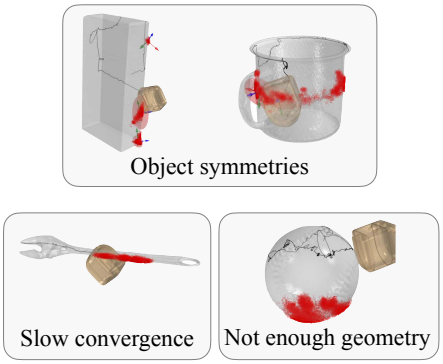

Figure 11: Failure modes on real-world trials: **(i)** it may be hard to converge to the true hypothesis for objects with symmetries, **(ii)** slow convergence of the filter can lead to large pose uncertainty, **(iii)** lack of discernible geometry can result in drift from the true mode.

Limitations: The current formulation is limited to a moving sensor relative to a fixed-pose object (or vice-versa). In future work, we wish track a dynamic object in the filter's state through **(i)** visual measurements [22], and/or **(ii)** local in-hand tracking [16]. Our method currently does not work in scenarios where we lack ground-truth object models. Along with reconstructing objects [65], 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 [66]. Object compositionality doesn't hold for deformable or articulated objects, so modeling these properties is an interesting future direction [67]. With a differentiable filter [68] we can fine-tune the system end-to-end for more robust real-world performance.

References

[1] J. Heller. *Catch-22: a novel*, volume 4. Simon and Schuster, 1961.

[2] J. Shi, J. Z. Woodruff, P. B. Umbanhowar, and K. M. Lynch. Dynamic in-hand sliding manipulation. *IEEE Trans. on Robotics (TRO)*, 33(4):778–795, 2017.

[3] N. Chavan-Dafle, R. Holladay, and A. Rodriguez. In-hand manipulation via motion cones. *Proc. Robotics: Science and Systems (RSS)*, 2018.

[4] Y. She, S. Wang, S. Dong, N. Sunil, A. Rodriguez, and E. Adelson. Cable manipulation with a tactile-reactive gripper. *Intl. J. of Robotics Research (IJRR)*, 40(12-14):1385–1401, 2021.

[5] O. M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, et al. Learning dexterous in-hand manipulation. *Intl. J. of Robotics Research (IJRR)*, 39(1):3–20, 2020.

[6] T. Chen, J. Xu, and P. Agrawal. A system for general in-hand object re-orientation. In *Proc. Conf. on Robot Learning (CoRL)*, pages 297–307. PMLR, 2022.

[7] B. Sundaralingam and T. 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)*, pages 231–238. IEEE, 2018.

[8] S. Brahmbhatt, A. Handa, J. Hays, and D. Fox. Contactgrasp: Functional multi-finger grasp synthesis from contact. In *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 2386–2393. IEEE, 2019.

[9] W. Yuan, S. Dong, and E. H. Adelson. GelSight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12):2762, 2017.

[10] E. Donlon, S. Dong, M. Liu, J. Li, E. Adelson, and A. Rodriguez. GelSlim: A high-resolution, compact, robust, and calibrated tactile-sensing finger. In *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 1927–1934. IEEE, 2018.

[11] M. Lambeta, P.-W. Chou, S. Tian, B. Yang, B. Maloon, V. R. Most, D. Stroud, R. Santos, A. Byagowi, G. Kammerer, et al. DIGIT: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation. *IEEE Robotics and Automation Letters (RA-L)*, 5(3):3838–3845, 2020.

[12] R. Li, R. Platt, W. Yuan, A. ten Pas, N. Roscup, M. A. Srinivasan, and E. Adelson. Localization and manipulation of small parts using GelSight tactile sensing. In *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 3988–3993. IEEE, 2014.

[13] M. Bauza, E. Valls, B. Lim, T. Sechopoulos, and A. Rodriguez. Tactile object pose estimation from the first touch with geometric contact rendering. In *Proc. Conf. on Robot Learning, CoRL*, 2020.

[14] M. Bauza, A. Bronars, and A. Rodriguez. Tac2Pose: Tactile object pose estimation from the first touch. *Intl. J. of Robotics Research (IJRR)*, 2022.

[15] P. Sodhi, M. Kaess, M. Mukadam, and S. Anderson. Learning tactile models for factor graph-based estimation. In *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 13686–13692. IEEE, 2021.

[16] P. Sodhi, M. Kaess, M. Mukadam, and S. Anderson. Patchgraph: In-hand tactile tracking with learned surface normals. In *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2022.

[17] J. Hawkins, M. Lewis, M. Klukas, S. Purdy, and S. Ahmad. A framework for intelligence and cortical function based on grid cells in the neocortex. *Frontiers in neural circuits*, 12:121, 2019.

[18] S. J. Lederman and R. L. Klatzky. Hand movements: A window into haptic object recognition. *Cognitive psychology*, 19(3):342–368, 1987.

[19] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte Carlo localization for mobile robots. In *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*, volume 2, pages 1322–1328. IEEE, 1999.

[20] B. Calli, A. Singh, J. Bruce, A. Walsman, K. Konolige, S. Srinivasa, P. Abbeel, and A. M. Dollar. Yale-CMU-Berkeley dataset for robotic manipulation research. *Intl. J. of Robotics Research (IJRR)*, 36(3):261–268, 2017.

[21] L. Downs, A. Francis, N. Koenig, B. Kinman, R. Hickman, K. Reymann, T. B. McHugh, and V. Vanhoucke. Google scanned objects: A high-quality dataset of 3D scanned household items. *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2022.

[22] X. Deng, A. Mousavian, Y. Xiang, F. Xia, T. Bretl, and D. Fox. PoseRBPF: A Rao–Blackwellized particle filter for 6D object pose tracking. *IEEE Trans. on Robotics (TRO)*, 37(5):1328–1342, 2021.

[23] A. Yamaguchi and C. 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)*, pages 1045–1051. IEEE, 2016.

[24] B. Ward-Cherrier, N. Pestell, L. Cramphorn, B. Winstone, M. E. Giannaccini, J. Rossiter, and N. F. Lepora. The TacTip family: Soft optical tactile sensors with 3D-printed biomimetic morphologies. *Soft robotics*, 5(2):216–227, 2018.

[25] A. Alspach, K. Hashimoto, N. Kuppuswamy, and R. Tedrake. Soft-bubble: A highly compliant dense geometry tactile sensor for robot manipulation. In *Proc. IEEE Intl. Conf. on Soft Robotics (RoboSoft)*, pages 597–604. IEEE, 2019.

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

[27] S. Wang, Y. She, B. Romero, and E. Adelson. GelSight Wedge: Measuring high-resolution 3D contact geometry with a compact robot finger. In *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE, 2021.

[28] S. Wang, M. M. Lambeta, P.-W. Chou, and R. Calandra. TACTO: A fast, flexible, and open-source simulator for high-resolution vision-based tactile sensors. *IEEE Robotics and Automation Letters (RA-L)*, 2022.

[29] A. Agarwal, T. Man, and W. Yuan. Simulation of vision-based tactile sensors using physics based rendering. In *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 1–7. IEEE, 2021.

[30] Z. Si and W. Yuan. Taxim: An example-based simulation model for gelsight tactile sensors. *IEEE Robotics and Automation Letters (RA-L)*, 2022.

[31] C. Choy, J. Park, and V. Koltun. Fully convolutional geometric features. In *Proc. Intl. Conf. on Computer Vision (ICCV)*, pages 8958–8966, 2019.

[32] M. A. Uy and G. H. Lee. PointNetVLAD: Deep point cloud based retrieval for large-scale place recognition. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 4470–4479, 2018.

[33] J. Komorowski. MinkLoc3D: Point cloud based large-scale place recognition. In *Proc. Winter Conf. on Applications of Computer Vision (WACV)*, pages 1790–1799, 2021.

[34] C. Corcoran and R. Platt. Tracking object pose and shape during robot manipulation based on tactile information. In *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*, volume 2. Citeseer, 2010.

[35] A. Petrovskaya and O. Khatib. Global localization of objects via touch. *IEEE Trans. on Robotics (TRO)*, 27(3):569–585, 2011.

[36] L. Zhang and J. 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*, pages 3805–3812. IEEE, 2012.

[37] M. Chalon, J. Reinecke, and M. Pfanne. Online in-hand object localization. In *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 2977–2984. IEEE, 2013.

[38] B. Saund, S. Chen, and R. Simmons. Touch based localization of parts for high precision manufacturing. In *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 378–385. IEEE, 2017.

[39] M. C. Koval, M. Klingensmith, S. S. Srinivasa, N. S. Pollard, and M. Kaess. The manifold particle filter for state estimation on high-dimensional implicit manifolds. In *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 4673–4680. IEEE, 2017.

[40] S. Thrun. Particle filters in robotics. In *UAI*, volume 2, pages 511–518. Citeseer, 2002.

[41] Z. Pezzementi, C. Reyda, and G. D. Hager. Object mapping, recognition, and localization from tactile geometry. In *2011 IEEE International Conference on Robotics and Automation*, pages 5942–5948. IEEE, 2011.

[42] S. Luo, W. Mou, K. Althoefer, and H. Liu. Localizing the object contact through matching tactile features with visual map. In *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 3903–3908. IEEE, 2015.

[43] J. Bimbo, S. Luo, K. Althoefer, and H. Liu. In-hand object pose estimation using covariance-based tactile to geometry matching. *IEEE Robotics and Automation Letters (RA-L)*, 1(1): 570–577, 2016.

[44] K.-T. Yu and A. Rodriguez. Realtime state estimation with tactile and visual sensing. application to planar manipulation. In *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 7778–7785. IEEE, 2018.

[45] A. N. Chaudhury, T. Man, W. Yuan, and C. G. Atkeson. Using collocated vision and tactile sensors for visual servoing and localization. *IEEE Robotics and Automation Letters (RA-L)*, 7 (2):3427–3434, 2022.

[46] S. Dong and A. Rodriguez. Tactile-based insertion for dense box-packing. In *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 7953–7960. IEEE, 2019.

[47] S. Kim and A. Rodriguez. Active extrinsic contact sensing: Application to general peg-in-hole insertion. In *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE, 2022.

[48] T. Kelestemur, R. Platt, and T. Padir. Tactile pose estimation and policy learning for unknown object manipulation. *arXiv preprint arXiv:2203.10685*, 2022.

[49] R. Gao, Z. Si, Y.-Y. Chang, S. Clarke, J. Bohg, L. Fei-Fei, W. Yuan, and J. Wu. ObjectFolder 2.0: A multisensory object dataset for Sim2Real transfer. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[50] N. Kuppuswamy, A. Castro, C. Phillips-Grafflin, A. Alspach, and R. Tedrake. Fast model-based contact patch and pose estimation for highly deformable dense-geometry tactile sensors. *IEEE Robotics and Automation Letters (RA-L)*, 5(2):1811–1818, 2019.

[51] M. K. Johnson and E. H. Adelson. Retrographic sensing for the measurement of surface texture and shape. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1070–1077, 2009. doi:10.1109/CVPR.2009.5206534.

[52] M. K. Johnson, F. Cole, A. Raj, and E. H. Adelson. Microgeometry capture using an elastomeric sensor. *ACM Trans. Graph.*, 30(4), July 2011. ISSN 0730-0301. doi:10.1145/2010324.1964941. URL https://doi.org/10.1145/2010324.1964941.

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

[54] R. Ambrus, V. Guizilini, N. Kuppuswamy, A. Beaulieu, A. Gaidon, and A. Alspach. Monocular depth estimation for soft visuotactile sensors. In *Proc. IEEE Intl. Conf. on Soft Robotics (RoboSoft)*, 2021.

[55] S. Suresh, Z. Si, J. G. Mangelson, W. Yuan, and M. Kaess. ShapeMap 3-D: Efficient shape mapping through dense touch and vision. In *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Philadelphia, PA, USA, May 2022.

[56] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 652–660, 2017.

[57] C. Choy, J. Gwak, and S. Savarese. 4D spatio-temporal ConvNets: Minkowski convolutional neural networks. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3075–3084, 2019.

[58] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *Proc. Intl. Conf. on 3D Vision (3DV)*, pages 239–248. IEEE, 2016.

[59] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2117–2125, 2017.

[60] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[61] W. Yuan, R. Li, M. A. Srinivasan, and E. H. Adelson. Measurement of shear and slip with a GelSight tactile sensor. In *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 304–311. IEEE, 2015.

[62] S. Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3):52–57, 2002.

[63] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu. DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. *ACM Transactions on Database Systems (TODS)*, 42(3):1–21, 2017.

[64] F. L. Markley, Y. Cheng, J. L. Crassidis, and Y. Oshman. Averaging quaternions. *Journal of Guidance, Control, and Dynamics*, 30(4):1193–1197, 2007.

[65] S. Suresh, M. Bauza, K.-T. Yu, J. G. Mangelson, A. Rodriguez, and M. 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.

[66] K. Aquilina, D. A. Barton, and N. F. Lepora. Shear-invariant sliding contact perception with a soft tactile sensor. In *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 4283–4289. IEEE, 2019.

[67] Y. Wi, P. Florence, A. Zeng, and N. Fazeli. VIRDO: Visio-tactile implicit representations of deformable objects. In *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE, 2022.

[68] R. Jonschkowski, D. Rastogi, and O. Brock. Differentiable particle filters: End-to-end learning with algorithmic priors. In *Proc. Robotics: Science and Systems (RSS)*, 2018.