MANIPULATION

NeuralFeels with neural fields: Visuotactile perception for in-hand manipulation

Sudharshan Suresh[1,2]*, Haozhi Qi[2,3], Tingfan Wu[2], Taosha Fan[2], Luis Pineda[2], Mike Lambeta[2], Jitendra Malik[2,3], Mrinal Kalakrishnan[2], Roberto Calandra[4,5], Michael Kaess[1], Joseph Ortiz[2]†, Mustafa Mukadam[2]

To achieve human-level dexterity, robots must infer spatial awareness from multimodal sensing to reason over contact interactions. During in-hand manipulation of novel objects, such spatial awareness involves estimating the object's pose and shape. The status quo for in-hand perception primarily uses vision and is restricted to tracking a priori known objects. Moreover, visual occlusion of objects in hand is imminent during manipulation, preventing current systems from pushing beyond tasks without occlusion. We combined vision and touch sensing on a multifingered hand to estimate an object's 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 studied multimodal in-hand perception in simulation and the real world, interacting with different objects via a proprioception-driven policy. Our experiments showed final reconstruction *F* scores of 81% and average pose drifts of 4.7 millimeters, which was further reduced to 2.3 millimeters with known object models. In addition, we observed that, under heavy visual occlusion, we could achieve improvements in tracking up to 94% compared with vision-only methods. Our results demonstrate that touch, at the very least, refines and, at the very best, disambiguates visual estimates during in-hand manipulation. We release our evaluation dataset of 70 experiments, FeelSight, as a step toward benchmarking in this domain. Our neural representation driven by multimodal sensing can serve as a perception backbone toward advancing robot dexterity.

INTRODUCTION

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. Now, robots lack the cognition to replicate even a fraction of the mundane tasks we perform, a trend summarized by Moravec's paradox (*1*). For dexterity in unstructured environments, a robot must first understand its spatial relationship with respect to the manipulated object. As robots move out of instrumented labs and factories to cohabit our spaces, there is a need for generalizable spatial perception (*2*).

Robots need dexterity beyond pick and place; although grasping a hammer or screwdriver may be straightforward, tool use requires the ability to rotate and regrasp in hand. Specific to in-hand dexterity, knowledge of object pose and geometry is crucial for policy generalization (*3*–*6*). As opposed to end-to-end supervision (*7*–*10*), these methods require a persistent three-dimensional (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 (*5*). Furthermore, it is common for practitioners to sidestep the perception problem entirely, retrofitting objects and environments with fiducials (*3*, *4*). To further progress toward general dexterity, a missing piece 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-occlusion is imminent, like rotating (*11*), reorienting (*5*, *10*), and sliding (*12*, *13*). In addition, vision often fails in the real world because of poor illumination, limited range, transparency, and specularity. Touch provides a direct window into these dynamic interactions, and human cognitive studies have reinforced its complementarity with vision (*14*).

Researchers have made advances in tactile sensing for multifinger robots (*15*), most prominent being vision-based fingertip sensors (*16*–*23*) like GelSight and DIGIT. Progress in simulation (*24*) enables practitioners to learn tactile observation models that transfer to real-world interactions (*22*, *25*, *26*). 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, what is the best strategy for representing spatial information? Coordinate-based learning, formalized as neural fields (*27*), 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 (*28*–*30*). They are continuous representations that have several advantages over their discrete counterparts like point clouds, meshes, and voxel maps—differentiability, precise reconstructions, and memory efficiency. Although initially developed for offline training, lightweight signed distance field (SDF) models (*31*–*34*) have made online perception possible. The ease of imparting generative priors (*35*) and pretraining (*36*) make neural fields more adaptable than classical methods.

Researchers have used neural fields not only for continuous 3D quantities like SDFs and radiance (*28*, *29*, *36*) but also for pose

[1]Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA. [2]FAIR, Meta, Menlo Park, CA 94025, USA. [3]Department of Electrical Engineering and Computer Sciences, UC Berkeley, Berkeley, CA 94720, USA. [4]Institute of Artificial Intelligence, Technische Universität Dresden, 01062 Dresden, Germany. [5]Centre for Tactile Internet with Human-in-the-Loop (CeTI), 01062 Dresden, Germany.
*Corresponding author. Email: suddhus@gmail.com
†Present address: Google DeepMind, Mountain View, CA, USA.

estimation (*34*, *37*), planning (*38*), and latent physics (*39*). Neural fields have shown promise in robot manipulation for learning policies (*40*), object deformation (*41*), scene dynamics (*38*, *42*), data generation (*43*), and transparent object manipulation (*44*, *45*). However, online perception and optimization of multimodal data remain challenges.

The domain of our work—an intersection of simultaneous localization and mapping (SLAM) and manipulation—has been studied for more than 2 decades. A first exemplar is from Moll and Erdmann (*46*), who reconstructed the shape and motion of an object rolled between robot palms. The combination of vision and touch has been explored for reconstructing the shape of fixed objects (*26*, *47*–*52*), tracking known objects (*53*–*55*), and global localization of known objects (*56*, *57*). In full SLAM, tactile-only methods have been investigated for simple objects via planar pushing (*58*, *59*) and specialized rolling fingertips (*60*, *61*). Closest to our work is the visuotactile SLAM system by Zhao *et al.* (*62*), combining dense touch from a single finger with red-green-blue (RGB) images, but it does not address the challenging case of in-hand manipulation.

NeuralFeels is an online solution to localize and reconstruct object shape via in-hand manipulation. It builds on prior works to demonstrate full SLAM with a multifinger robot for a priori unknown objects and robust tracking of known objects. We used a dexterous hand (*63*) sensorized with commercial vision-based touch sensors (*20*) and a fixed RGB-D camera (Fig. 1). With a proprioception-driven policy (*11*), we explored the object's extents through in-hand rotation—using the SLAM solution to guide that the policy is not an explicit objective of our work. This falls in line with prior works in SLAM for manipulation (*52*, *55*, *57*, *62*) that focused on perception by isolating the evaluation from the manipulation task.

Here, we studied the role that vision and touch play in interactive perception, the effects of occlusion, and visual sensing noise. We presented our robot with a novel object, and it inferred and tracked its geometry through vision, touch, and proprioception. To evaluate our work, we collected a benchmark dataset of 70 in-hand rotation trials in both the real world and simulation, with ground-truth object meshes and tracking. Our results for novel objects show average reconstruction *F*-scores of 81% with pose drifts of just 4.7 mm, which were further reduced to 2.3 mm with known computer-aided design (CAD) models. Under heavy occlusion, we demonstrate up to 94% improvement in pose tracking compared with vision-only methods. Our combination of rich sensing and spatial perception requires minimal hardware compared with 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.

RESULTS

Our multifingered robot hand was presented with a novel object, placed randomly between its fingertips. We rotated the object in hand, through a proprioception-driven policy (*11*), which gave rise to a stream of visual and tactile signals. We combined 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 is illustrated in Fig. 2.

We evaluated NeuralFeels over simulated and real-world interactions, totaling 70 experiments over different object classes. First, we demonstrated SLAM results for novel objects and highlight some qualitative examples. Next, we demonstrated pose tracking with an a priori shape for the manipulated object. Last, we analyzed the role that touch plays in improving perception under occlusion and visual sensing noise. Movies S1 and S2 visualize representative neural SLAM results for the bell pepper and rubber duck objects, respectively. Movie S3 provides a longer narrated summary of our results and methodology.

Metrics and baseline

Pose and shape metrics

We used the symmetric average Euclidean distance (ADD-S) (*64*), henceforth referred to as the pose metric, to evaluate tracking error over time. The ADD metric is commonly used in manipulation (*64*–*67*) as a geometrically interpretable distance metric for pose error. It is computed by subsampling the ground-truth object mesh and averaging the Euclidean distance between the point sets in the estimated and ground-truth object pose frames. Rather than pairwise distance, ADD-S considers the closest point distance, which disambiguates symmetric objects (*64*).

For shape, we compared how accurate (precision) and complete (recall) the neural SDF is in comparison with the ground-truth mesh. The *F*-score, an established metric in the multiview reconstruction community (*68*, *69*), combines these two criteria into an interpretable [0–1] value. To compute this metric, henceforth referred to as the shape metric, we first subsampled the ground-truth and reconstructed meshes in their object-centric frame. Given a distance threshold, in our case $\tau = 5$ mm, the precision measured the percentage of reconstructed points within τ distance from the ground-truth points. Conversely, recall measured the percentage of ground-truth points within τ distance from the reconstructed points. The harmonic mean of these two quantities gave us the *F*-score, which jointly captured surface reconstruction

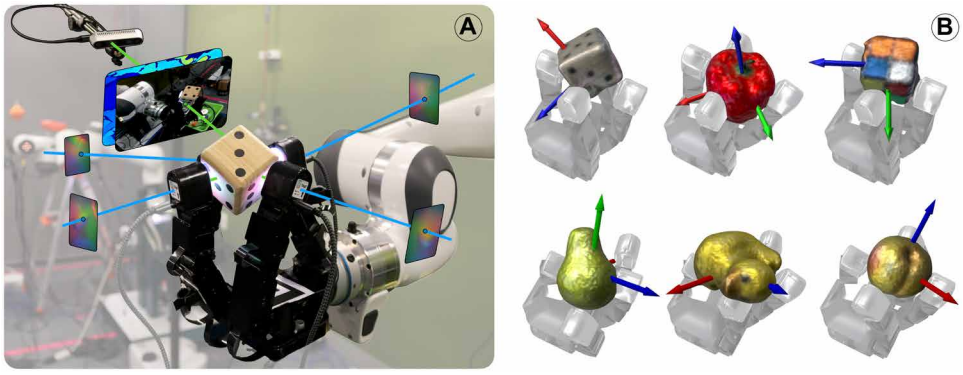


Fig. 1. Visuotactile perception with NeuralFeels. Our method estimates the pose and shape of novel objects during in-hand manipulation (**B**) by learning neural field models online from a stream of vision, touch, and proprioception (**A**).

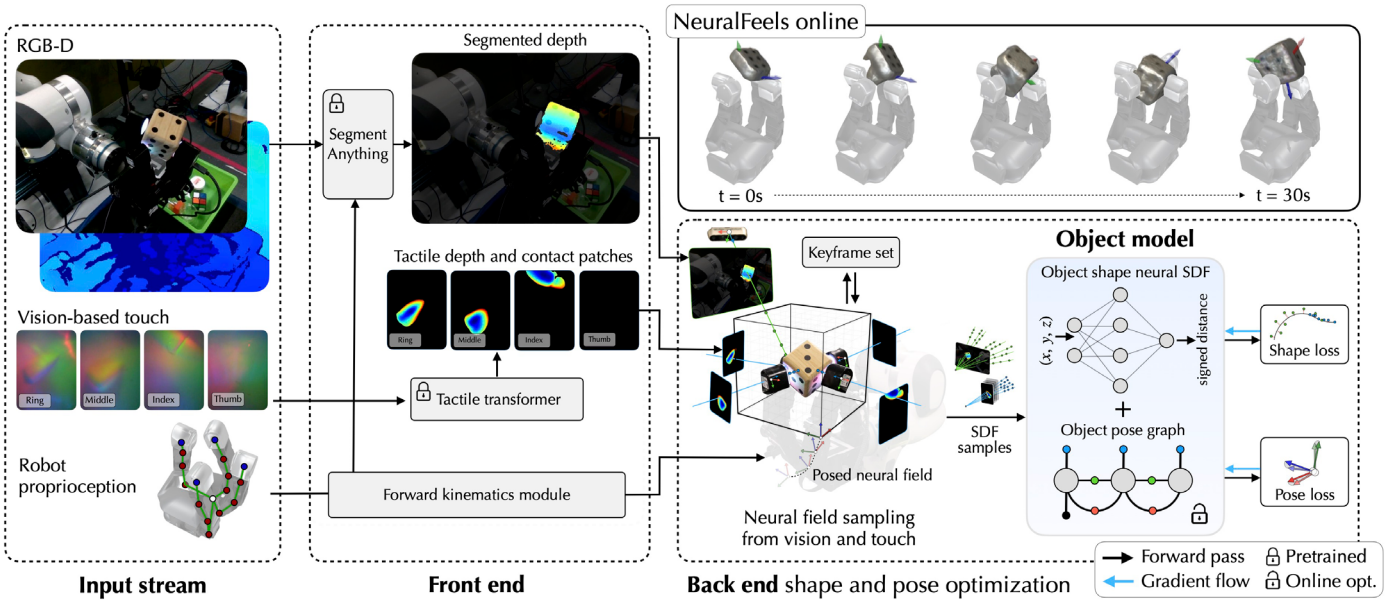


Fig. 2. A visuotactile perception stack amid interaction. An online representation of object shape and pose is built from vision, touch, and proprioception during in-hand manipulation. Raw sensor data are first fed into the front end, which extracts visuotactile depth with our pretrained models. Then, the back-end samples from the depth train a neural SDF, and the pose graph tracks the neural field.

accuracy and shape completion. Broadly, a higher F-score with tighter τ bounds implied better object reconstruction.

Ground-truth shape and pose

We evaluated these metrics against the ground-truth estimates of object shape and pose. For each object, the ground-truth shape was obtained from offline scans (fig. S1). The ground-truth object pose was straightforward in simulation experiments, directly exposed by Isaac Gym (*70*). In the real world, we estimated pseudo–ground truth via multicamera pose tracking of the experiment. Instrumented solutions, such as 3D motion capture, were infeasible given that they both visually and physically interfered with the experiments. We opted to install two additional cameras and ran NeuralFeels in pose-tracking mode with the ground-truth object shape. This represents the best tracking estimates given a known shape and occlusion-free vision. For further details, refer to the "Ground-truth shape and pose" section in the Supplementary Materials.

Object initialization

In practice, the object-centric reference frame in our SLAM experiments should be picked arbitrarily (such as the centroid of the initial point cloud or the robot fingers). However, the ground-truth reference frame was defined as the centroid of the complete CAD model, oriented along its major axis. This mismatch in the reference frames is expected in a causal system but will lead to an incorrect calculation of the object-centric shape metric. In addition, object tracking with a known shape is quite sensitive to initial orientation of the reference frame (*71*). To address these issues, we assumed that the initial object pose was known and aligned to the initial ground-truth pose. We instead focused on the subsequent tracking and shape reconstruction, which was challenging even with good initialization. In the future, a coarse initialization can be obtained from a feature-based front end (*72*). To ensure that our evaluation did not benefit from this object initialization, we only started computing our pose metric 5 s into each trial.

Neural SLAM: Object tracking and shape estimation

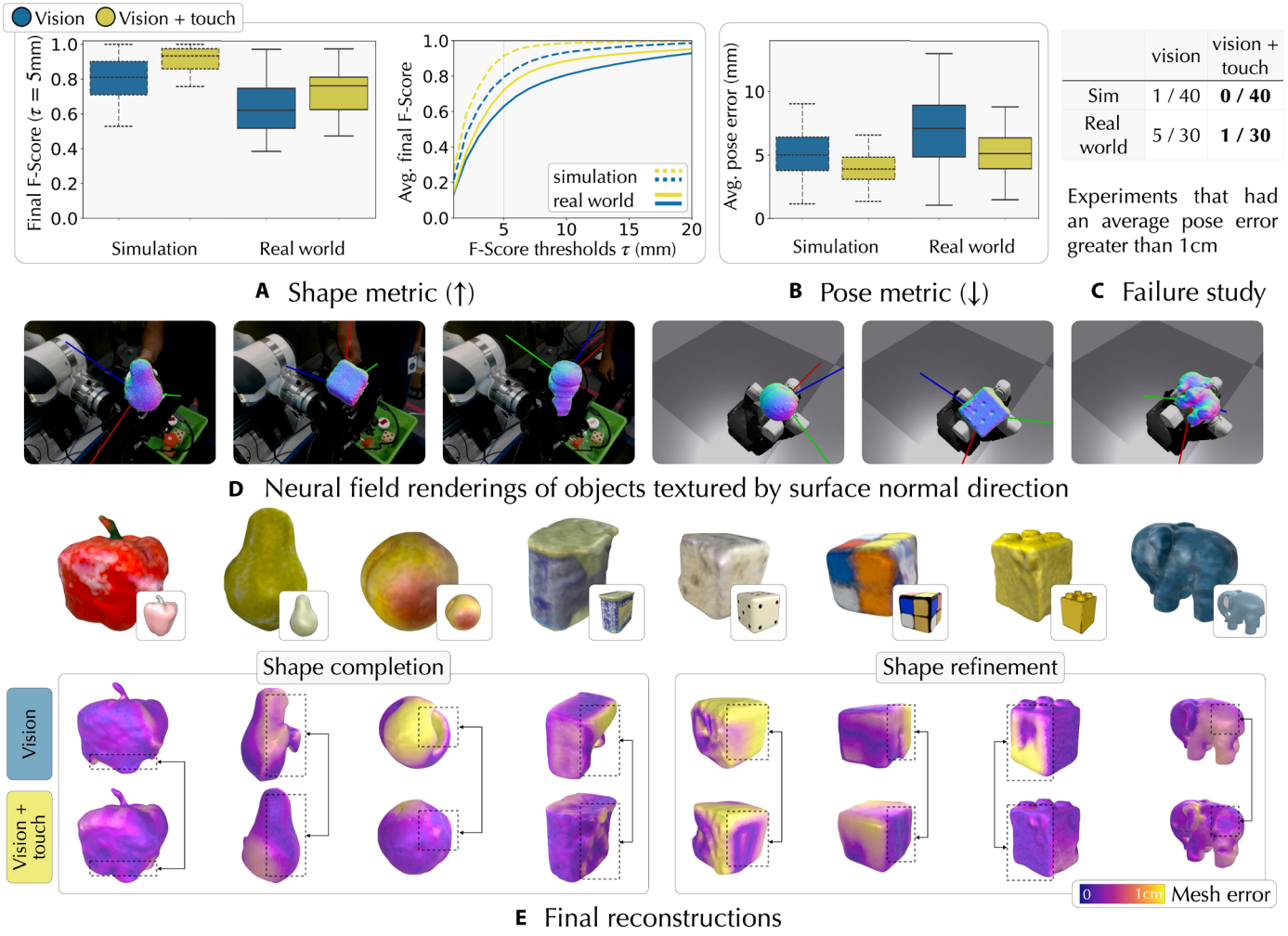
Motivation and importance

As a first experiment, we evaluated NeuralFeels' ability to track and reconstruct unknown objects from multimodal sensing. This is important for robots deployed in unstructured environments, such as households, with a priori unknown objects. We presented the robot with a novel object, and the robot was tasked with building an object model on the fly. Our SLAM method made no assumptions about the object geometry, which was built from scratch, or manipulation actions, which were decided at deployment. We processed visuotactile data sequentially without access to future information or category-level priors. This formulation aligns with prior dexterous manipulation works (*5*, *6*, *10*, *11*) and is less restrictive than that of Zhao *et al.* (*62*), where the object was always in contact with a single tactile sensor and the camera was unobstructed.

We evaluated more than a combined 70 experiments in simulation and the real world across 14 different objects. The objects were placed in hand, after which the policy collected 30 s of vision, touch, and proprioception data. Given that each run was nondeterministic, we averaged our results across five different seeds, resulting in a total of 350 trials. The first frame of each sequence only presented limited visual knowledge: a single side of a Rubik's cube or large die or the underside of a rubber duck. Through the course of any 30-s sequence, in-hand rotation exposed previously unseen geometries to vision, and touch filled in the rest of the occluded surfaces. In Fig. 3, we show the main set of results, where we compared the multimodal fusion schemes against ground truth.

Object reconstructions

Figure 3A shows the final shape metric at the end of each sequence for a fixed threshold τ. Here, we picked $\tau = 5$ mm for this evaluation, around 3% of the maximum diagonal length of the objects. The greater the value of the shape metric was, the closer the surface reconstructions were to ground truth. We observed large gains when



	vision	vision + touch
Sim	1 / 40	**0 / 40**
Real world	5 / 30	**1 / 30**

Experiments that had an average pose error greater than 1cm

Fig. 3. Summary of SLAM experiments. (**A** and **B**) Aggregated statistics for SLAM over a combined 70 experiments (40 in simulation and 30 in the real world), with each trial run over five different seeds. We compared across simulation and the real world to show low pose drift and high reconstruction accuracy. Each boxplot represents the aggregate error over all experiments, where the central line is the median, extents of the box are the upper and lower quartiles, and the whiskers represent 1.0× the IQR. (**C**) Number of trials that our method failed to track (and reconstruct) the object. (**D**) Representative examples of the final object pose and neural field renderings from the experiments. Each object was textured by mapping the surface normal directions to an RGB colormap. (**E**) Final 3D objects generated by marching cubes on our neural field. Here, we highlight the role touch played in both shape completion and shape refinement.

incorporating touch, with surface reconstructions on average 15.3% better in simulation ($P < 0.001$) and 14.6% better in the real world ($P < 0.001$). Our final reconstructions, as seen in Fig. 3E, had a median error of 2.1 mm in simulation and 3.9 mm in the real world. In addition, the second plot compares the final shape metrics against a range of τ thresholds. Here, we observed that multimodal fusion led to consistently better shape metrics across all τ values in simulation and the real world.

Object pose drift

In SLAM, there is a strong correlation between a low shape metric and high pose metric, given that one often leads to the other. Figure 3B plots the drift of the object's estimated pose with respect to the ground truth, with a lower drift being more accurate. We observed better tracking with respect to the vision-only baseline, with improvements of 21.3% in simulation ($P < 0.001$) and 26.6% in the real world ($P < 0.001$). Figure 3C reports the number of failures in vision-only tracking compared with NeuralFeels. Here, a failed experiment was defined as when the average pose drift exceeded a threshold of 10 mm. This was loosely based on Bauza *et al.* (*73*), where they considered 10 mm as a coarse initialization for tactile localization. To highlight the importance of our neural field, fig. S19 shows that our method outperformed a baseline that relied only on iterative closest point (ICP) frame-to-frame constraints.

Empirically, we observed a large pose drift in the first few seconds from initialization at ground truth because of an unknown shape. Over time, we built a better shape model, resulting in more accurate pose tracking (figs. S17 and S18). However, with pose regularization and the lack of long-term loop closures (*72*, *74*), small errors in pose would accumulate over time. This cascading effect is common in SLAM (*75*), where pose errors cause a disagreement between the reconstructed map and the physical world.

Because of this, we identified whether any trials had an average shape metric that deteriorated over time. This was done by computing the difference between the average shape metric across the first

50% of the sequences and the last 50% of the sequences. We concluded that $^{25}/_{150}$ (16%) of the real-world trials had a shape estimate that deteriorated over time and that the other $^{125}/_{150}$ (83%) improved. In simulation, our method performed better: $^{9}/_{200}$ (4.5%) trials had shape estimates that deteriorated over time, and $^{191}/_{200}$ (95%) improved.

Qualitative results

Figure 3D 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 (*76*), are shown in Fig. 3E alongside the ground-truth meshes. Below that, we highlight the gains with visuotactile integration for each of the reconstructed objects. We categorize these into shape completion—coverage of object surfaces that were occluded from vision—and shape refinement—touch measurements that complemented vision to better reconstruct visible surfaces.

Figure 4 shows incremental pose tracking and reconstruction of objects across different time slices of a few representative experiments. At each time step, we highlight the input stream, front-end depth, and output object model. Movies S1 and S2 provide an animated version of the experiments in Fig. 4A. In the current formulation of our problem, touch-only SLAM was not permissible. This is because the tracking (and thereby reconstruction) failed early in the sequence because of a lack of prior shape information and the field of view of the sensor could not rapidly give us global geometry.

Object tracking given an a priori known shape

Motivation and importance

These experiments studied the accuracy of pose tracking with NeuralFeels when provided with a priori known object CAD models. Tracking known geometries is an active area of research in manipulation (*5*, *71*), with some works that incorporate touch as well (*13*, *53*–*55*, *77*). This is applicable in environments like warehouses and manufacturing lines, where robots have intimate knowledge of the manipulated objects (*77*). It is further useful in household scenarios, where the robot has already generated an object model through interaction.

In implementation, the object's SDF was precomputed from a given CAD model. During the runtime, we froze the weights of the neural field and only performed visuotactile tracking with the front-end estimates. Similar to the SLAM experiments, we ran each of the 70 experiments over five seeds and report the pose metrics with respect to ground truth.

Results from pose tracking

Figure 5A 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 metrics over the sequences are plotted in Fig. 5B. We observed low, bounded pose error even with imprecise visual segmentation (fig. S24) and sparse touch signals. In Fig. 5C, we observed 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 observed that incorporating touch could refine our pose estimates, with a decrease in average pose error by 22.29% in simulation ($P < 0.001$) and 3.9% in the real world ($P = 0.21$). We posit that the relatively high real-world P value is because the real DIGIT elastomer was less sensitive, leading to sparser contacts. Sparse contacts played a large role in full SLAM, by coarsely reconstructing unseen surfaces, but they only played a refinement role when the full shape was known. In addition, the viewpoint did not have many occlusions—in the following section, we highlight greater improvements when visual sensing was suboptimal.

Perceiving under duress: Occlusion and visual depth noise

Motivation and importance

In this section, we explore the broader benefits of fusing touch and vision in challenging scenarios—occlusion and visual noise. The previous results were achieved through largely favorable camera positioning and precise stereo depth tuning. This attention to detail was necessary for prior practitioners as well (*5*, *10*), but could we also use touch to improve over suboptimal visual data? We considered two such scenarios in simulation, where we could freely control these parameters, and evaluated 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 would 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 considered tracking the pose of a known Rubik's cube. We simulated 200 different cameras in a sphere of radius 0.5 m, each facing toward the robot. As shown in Fig. 6A, each camera captured a unique vantage point of the same in-hand sequence, with varying levels of robot-object occlusion. This served as a proxy for occlusion faced by an egocentric or fixed camera when either the hand or environment occluded the object.

To simplify the experiment, we assumed the upper-bound performance of the vision-only front end by providing ground-truth object segmentation masks. We characterized 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 corresponded to viewpoints beneath the hand (most occluded), and those closer to 1 corresponded to cameras placed atop (least occluded). We ran pose tracking experiments for each of the 200 cameras in two modes, vision-only and visuotactile, and compared them.

In Fig. 6A, we colormapped each camera view on the basis of the pose-tracking improvements from incorporating touch. On average, the improvement across all cameras was 21.2%, and it peaked at 94.1% at heavily occluded views. Across the [0−1] range of occlusion scores, we had $P < 0.001$. We inset frames from a few representative viewpoints and their corresponding relative improvement with visuotactile fusion. In Fig. 6B, the pose error for each modality is further plotted versus the [0−1] occlusion score. This corroborated the idea that touch refined perception in low-occlusion regimes and robustified it in high-occlusion regimes.

The effects of noisy visual depth

The depth from commodity RGB-D sensors is degraded as a function of camera-robot distance, environment lighting, and object specularity. Even in ideal scenarios, the RealSense depth algorithm has 35 hyperparameters (*78*) that considerably affect the front-end input to NeuralFeels. To simulate this, we corrupted the depth maps progressively with realistic RGB-D noise and observed the tracking performance for a known geometry.

As implemented by Handa *et al.* (*79*), we simulated common sources of depth-map errors as a sequence of pixel shuffling,

quantization, and high-frequency noise. The depth noise factor D determined the magnitude of these operations, with the depth maps visualized in Fig. 6C. All prior simulation experiments had been collected with $D = 5$, but here, we varied the magnitude from 0 to 50 in intervals of 10. At each noise level, we ran pose tracking across the five Rubik's cube experiments with five unique seeds, resulting in a total of 150 experiments. In Fig. 6C, we plotted error against the noise factor D, showing an expected upward trend in error with noise. However, we saw better tracking when fusing touch, especially in high-noise regimes.

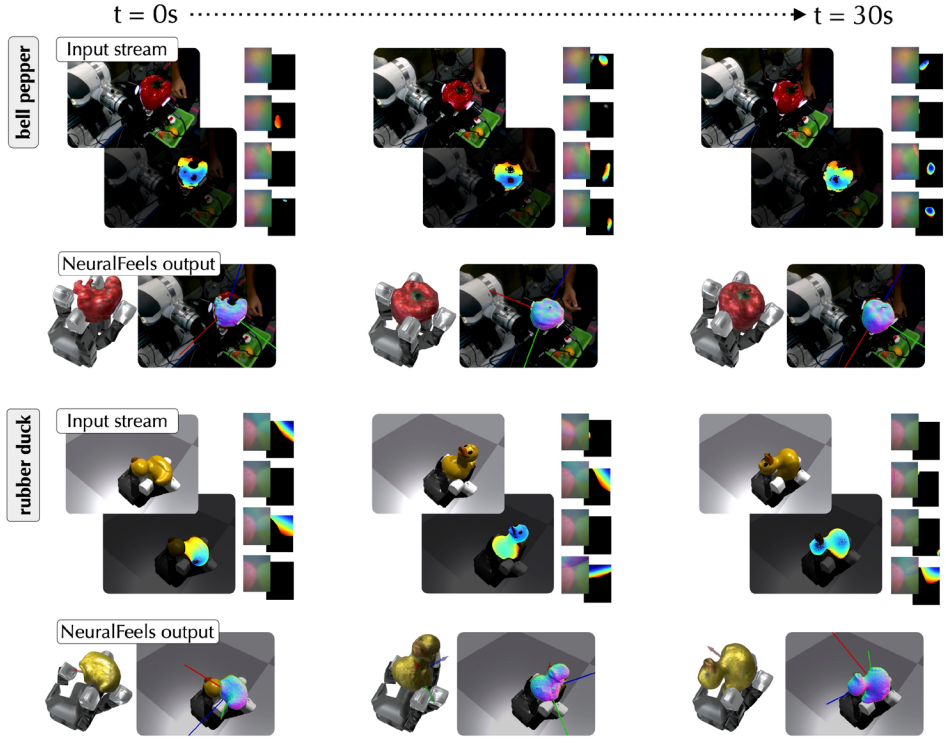
DISCUSSION

The experiments show that NeuralFeels achieves robust object-centric SLAM for multimodal, multifinger manipulation. As shown in Fig. 3A, we achieved an average reconstruction F-score of 81% across simulation and real-world experiments on novel objects. Simultaneously, we stably tracked these objects amid interaction with minimal drift, an average of 4.7 mm. Although 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 (*26*, *77*, *80*), now applied on a dexterous manipulation platform.

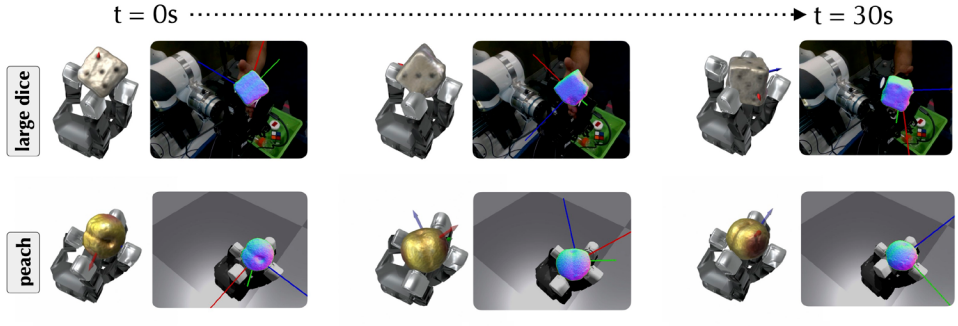
Interactive perception is far from ideal; an embodiment can more often than not get in the way of sensing. As seen in Fig. 4, in-hand manipulation suffers from challenges such as frequent occlusions, limited field of view, noisy segmentation, and rapid object motion. Proprioception helps focus the perception problem: We can accurately singulate the object of interest through embodied prompting (refer to the "Front end" section of Materials and Methods). 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.%/26.6% in pose tracking.

Qualitatively, we see that touch performs two key functions: disambiguating noisy front-end estimates and providing context in the presence of occlusion. The former alleviates the effect of noisy visual segmentation and depth with colocated local information for mapping and localization. The latter provides important context hidden from visual sensing, like the occluded face of the large die or back of the rubber duck. The final reconstructions in Fig. 3E support these findings, with improved shape completion and refinement.

With a known shape ("Object tracking given an a priori known shape" section of Results), touch plays a refinement role (Fig. 5) when there are not many visual occlusions. The largest gains from incorporating touch, expectedly, are in heavy-occlusion regimes (Fig. 6, A and B), where we observed up to 94.1% improvements at certain camera viewpoints. To our knowledge, a detailed study on how object visibility affects perception has not been explored in prior manipulation works. This demonstrates not just the complementary nature of the



A Representative SLAM results from the bell pepper (real world) and rubber duck (simulation) objects, alongside input RGB-D and tactile images at each time step



B Representative SLAM results at each time step from the large dice (real world) and peach (simulation)

Fig. 4. Representative SLAM results. (**A**) We show the input stream of RGB-D and tactile images, paired with the posed reconstruction at time step t for the bell pepper and rubber duck objects. In each case, we partially reconstructed the object at the initial frame and built the surfaces out progressively over each 30-s experiment. The 3D visualizations were generated by marching cubes, in addition to the rendered normals of the neural field projected onto the visual image. The rendering was textured by mapping the surface normal directions to an RGB colormap. (**B**) Further representative results with the large dice (real-world) and peach (simulation) objects.

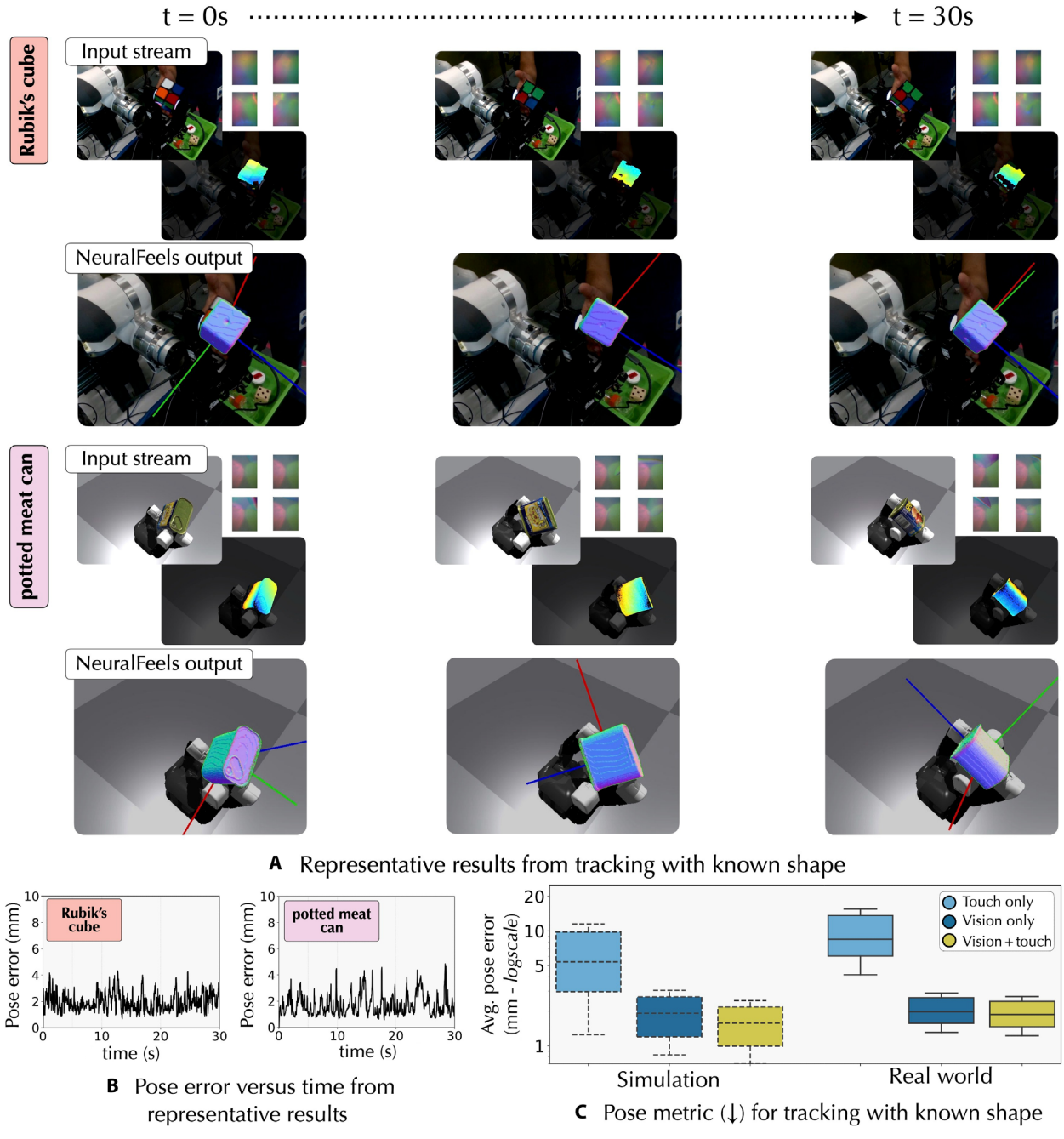
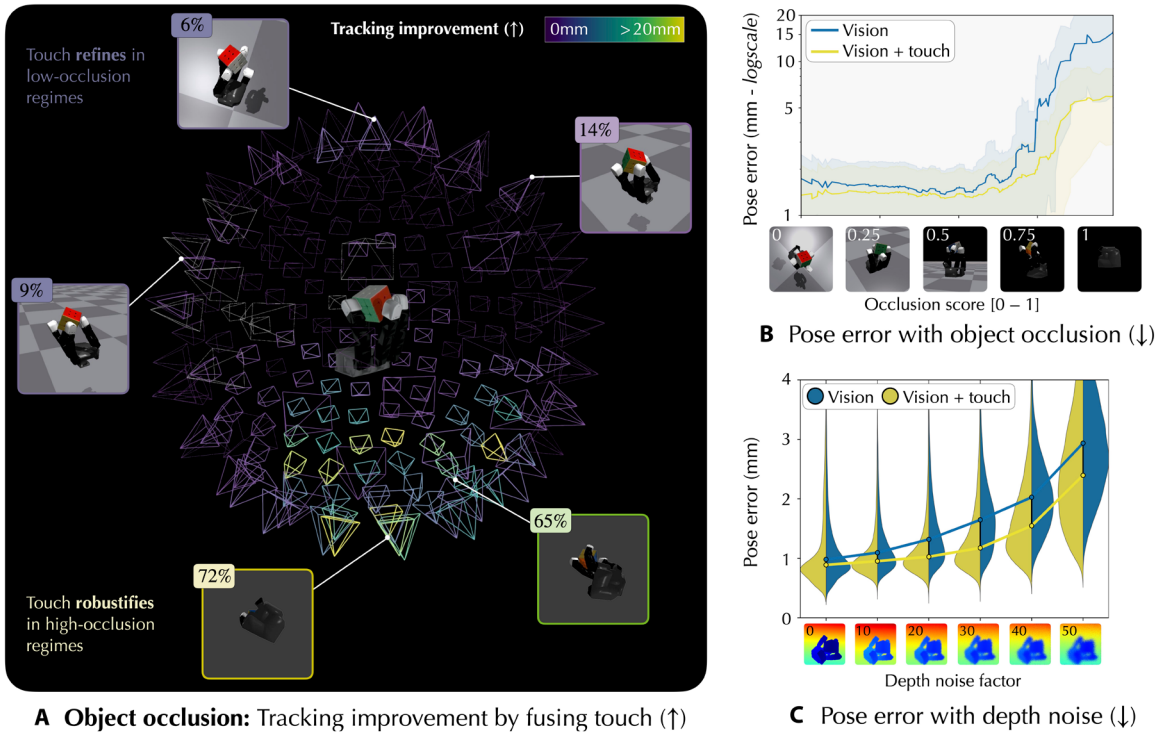


Fig. 5. Neural pose tracking of known objects. (**A**) We show the input stream of RGB-D and tactile images paired with the pose tracking at time step *t* for the Rubik's cube and potted meat can objects. With a known ground-truth shape, we could robustly track objects with vision and touch. Each experiment was 30 s long, and the object renderings were textured by mapping the surface normal directions to an RGB colormap. (**B**) We observed reliable tracking performance, with average pose errors of 2 mm through the sequence. (**C**) Aggregated statistics for pose tracking over a combined 70 experiments (40 in simulation and 30 in the real world), with each trial run over five different seeds. Each boxplot represents the aggregate pose error in log scale, where the central line is the median, extents of the box are the upper and lower quartiles, and the whiskers represent 0.25× the IQR. With a known object model and good visibility, touch played the role of pose refinement. In addition, we note that touch-only tracking is error prone and infeasible.

modalities but, further, the ideal configurations for occlusion-free manipulation. Last, our results in tactile-only tracking (Fig. 5C) support the findings of Smith *et al.* (*49*) that learning exclusively from touch leads to poor performance because it lacks any global context.

As opposed to an end-to-end perception, our modular stack marries pretraining with online learning. This allows us to combine foundation models trained on large-scale image and tactile data (front end) with SLAM as online learning (back end). Furthermore, our back end is a combination of state-of-the-art neural models (*29*)



A **Object occlusion:** Tracking improvement by fusing touch (↑)

B Pose error with object occlusion (↓)

C Pose error with depth noise (↓)

Fig. 6. Ablations on occlusions and sensing noise. (**A**) Pose tracking results from 200 simulated cameras in a sphere of radius 0.5 m, each facing toward the robot. Each camera view is colormapped on the basis of the pose tracking improvements from incorporating touch, when compared against vision only. At occlusion-heavy points of view, visuotactile fusion provided an unobstructed local perspective leading to improved tracking performance. (**B**) We computed a [0–1] occlusion score for each of the 200 experiments and plotted the pose errors against it. We observed that touch played a larger role when vision was heavily occluded and a refinement role when there was negligible occlusion. The shaded regions represent 1 SD from the mean. (**C**) We simulated noise in visual depth measurements and plotted the error distribution against the depth noise factor *D* as a violin plot. The inset image shows the qualitative depth noise for each *D*, and the inner markers represent the median pose error. We observed that, with an increase in noise, adding touch led to lower error distribution.

with classical least-squares optimization (*81*) that have found success in SLAM (*82*). This modular design has benefits for future generalization of our system: Other models of tactile sensors (*16*, *19*, *22*) can be easily integrated as long as they can be accurately simulated; alternate scene representations (*83*, *84*) can supplant our neural field model; additional state knowledge can be integrated as factor graph costs like tactile odometry (*62*) and force constraints (*59*); and any combination of tactile and visual sensors can be fused given appropriate calibration and kinematics.

NeuralFeels is relevant to researchers who require spatial perception with a single camera and affordable tactile sensing. It can be extended to not only in-hand rotation but also object-centric tasks like reorientation (*10*), pick and place (*77*), insertion (*61*), nonprehensile sliding (*85*), and planar pushing (*59*). Although not explored in this work, the benefit of an online SDF is the ability to seamlessly plan for dexterous interactions. Recent works demonstrate the benefit of a priori known object point clouds (*6*) and SDFs (*86*) for goal-conditioned planning, and running our perception stack in the loop is the next natural step.

System limitations

Our findings indicate that the benefits of multimodal fusion are less pronounced in real-world deployment when compared with simulation. This is a common problem in sim-to-real manipulation—prior works have encountered similar disparities in object pose estimation (*3*, *5*). In addition, we identify that the DIGIT elastomer is less sensitive in real-world deployment, leading to sparser contacts (Fig. 4); our reinforcement learning (RL) policy is less reliable in the real world, often requiring human intervention and causing large jumps in object motion (fig. S21). To tackle these shortcomings, we can focus on real-world fine-tuning of our simulator (*87*) and explicitly modeling sensor deformation and stress (*88*). Through multimodal RL (*6*, *10*), we can deliver more robust policies than those driven "blindly" by proprioception.

We are currently restricted to a fixed-camera setup, with an online hand-eye calibration or egocentric vision; this can be relaxed. Depth uncertainty (*89*) is valuable information for our neural model to handle visually adversarial objects like glass and metal. We used vision-based touch (*20*) over tactile arrays (*90*) or binary sensing (*7*), but future work can consider the merits of each. In the section titled "The role of touch" found in the Supplementary Materials, we present ablations on the benefits of higher resolution and a comparison against binary sensing. In our SLAM experiments, each pose graph iteration takes 0.79 ± 0.36 s [20 iterations of Levenberg-Marquardt (LM) (*75*)], and the shape optimization takes 0.06 ± 0.09 s (one iteration of gradient descent). For execution in a real-time loop, we can speed up Segment Anything Model (SAM) inference time (*91*), reduce SDF samples and downsample feature grid resolution, and substitute the pose graph with an incremental optimizer (*92*). Last, we can increase tracking robustness through feature-based methods (*93*) and loop-closure detection (*72*).

Future directions

Our method learns the 3D geometry of a novel object from scratch, and thus the pose tracker has a higher chance of failure in the initial seconds, when the SDF is unknown. In addition, our rotation policy might not completely explore the object in the real world, resulting in shape metrics that are lower than those seen in simulation. Given an initial occluded view, integration of large reconstruction models (*36*, *94*, *95*) can yield a good initial-guess SDF. In manipulation, Wang *et al.* (*48*) have seen promising results in using shape priors for visuotactile reconstruction of fixed objects.

Geometry is just a starting point for neural models: Interaction reveals latent properties like texture (*85*), friction (*39*), and object dynamics (*96*). With neural fields, we can embed these latents as auxiliary optimization variables to benefit tasks that go beyond just spatial quantities. Applications can range from learning to manipulate inertially challenging objects (like a hammer) to identifying a grasp point from local texture (like a saucepan handle).

In summary, NeuralFeels leverages vision, touch, and robot proprioception to reconstruct and track novel objects with high precision. The system is simpler than complex fiducial tracking, uses affordable touch sensing, and provides more interpretable output than end-to-end perception. Our approach combines ideas from SLAM, neural rendering, and tactile simulation and serves as an important step toward advancing robot dexterity.

MATERIALS AND METHODS

Similar to classical SLAM frameworks, NeuralFeels first has a front end, which converted the vision (RGB-D) and touch (RGB) input stream into a format suitable for estimation (segmented depth). Thereafter, the back end fused these data into an optimization structure that inferred the object model: an evolving posed object SDF. An illustration of the entire pipeline is found in Fig. 2, which we refer the reader back to throughout this section. In addition, a narrated summary of our method can be found in movie S3.

Task definition

NeuralFeels incrementally built an object model, simultaneously optimizing for the object SDF network's weights θ and its corresponding pose x_t at time step t. For object exploration, we used a proprioception-driven policy π_t that executed the optimal action to achieve stable rotation. The input stream (Fig. 2) of all sensors $\mathcal{S}$ consisted of the following: RGB-D vision—image I_t^c and depth D_t^c from a calibrated camera $c \in \mathcal{S}$; RGB touch—images I_t^s from four DIGITs (*20*); $s \in \{d_{\text{index}}, d_{\text{middle}}, d_{\text{ring}}, d_{\text{thumb}}\} \in \mathcal{S}$; and proprioception—joint angles $\mathbf{q}_t$ from robot encoders.

Robot hardware and simulation

The Allegro Hand (*63*) was retrofit with four DIGIT vision-based tactile sensors (*20*) at each of the distal ends. The DIGIT produced a 240 pixel–by–320 pixel RGB image of the physical interaction at 30 Hz. The Allegro published 16D joint angles so as to situate the tactile sensors with respect to the base frame. The hand was rigidly mounted on a Franka Panda arm, with an Intel RealSense D435 RGB-D camera placed at approximately 27 cm from its palm. The camera extrinsics were computed with respect to the base frame of the Allegro through ArUco (*97*) hand-eye calibration. For our vision pseudo–ground truth, we used three such cameras in the workspace (Fig. 7), jointly calibrated via Kalibr (*98*), to achieve approximately 1-pixel reprojection error. Our simulator replicated the real-world setup: a combination of the Isaac Gym physics simulator (*70*) with the TACTO touch renderer (*24*). In this case, we recorded and stored the true ground-truth object pose directly from Isaac Gym.

FeelSight: A visuotactile perception dataset

Visuotactile perception lacks a dataset that has driven progress in adjacent fields like visual tracking (*99*), SLAM (*100*), and RL (*101*). Toward this, we collected our FeelSight dataset for visuotactile manipulation. We used an in-hand rotation policy to collect vision, touch, and proprioception for 30 s per trial.

RL for object rotation

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 multifingered hand, in-hand rotation, is an ideal choice for the interactive perception problem. We adopted the method of Qi *et al.* (*11*) wherein they trained a proprioception-based policy in simulation and directly transferred it to the real world. The policy training and deployment, reward function, and performance are discussed in the "In-hand rotation policy" section of the Supplementary Materials. In all our experiments, a single policy π_t updated at 20 Hz (300-Hz low-level PD control) via the Robot Operating System (ROS) Allegro package.

This achieved multifingered rotation of novel objects and interesting visuotactile stimuli. The dataset has five rotation trials each of six objects in the real world and eight objects in simulation for a total of 35 min of interaction. As explained in Fig. 7, we recorded a pseudo–ground truth in the real world and exact ground-truth poses in simulation. The policy resulted in a translation/rotation of 25 mm per s/32.6° per s in simulation and 20 mm per sec/9.9° per sec in the real world.

The selected objects varied in geometry and size from 6 to 18 cm in diagonal length. Empirically, objects with irregular aspect ratios were harder to manipulate with the hand morphology; our choice of objects was based on the ability of our RL policy rather than the SLAM solution. Deformable object manipulation was deemed out of scope because we relied on Isaac Gym (*70*) and TACTO (*24*), which assumed rigid body simulation. Ground-truth real-world meshes were created with the Revopoint 3D scanner (*102*), and the simulated objects used ground-truth meshes from the Yale-CMU-Berkeley (YCB) (*103*) and ContactDB (*104*) datasets.

For objects like the Rubik's cube, we assisted the policy through human intervention in case of slip events (fig. S21). In the real world, we found that, with this robot hand morphology, it was difficult to achieve gaits for stable cube rotation. This was because, unlike a prior work that pivots the object atop the fingers using gravity (*11*), we relied on frictional contact to get tactile signals on our lateral-facing DIGITs. Thus, we opted for this strategy of derisking our experiments with a human in the loop. These interventions enabled us to collect a large set of experiments, but they were adversarial to perception given that they led to additional occlusions and sudden jumps in object pose.

Method overview

We represented the object SDF as a neural network with weights θ, whose output was transformed by the current object pose x_t. This continuous function $F_{x_t}^{\theta}(\mathbf{p})$: $\mathbb{R}^3 \rightarrow \mathbb{R}$ mapped a 3D coordinate $\mathbf{p}$ to a

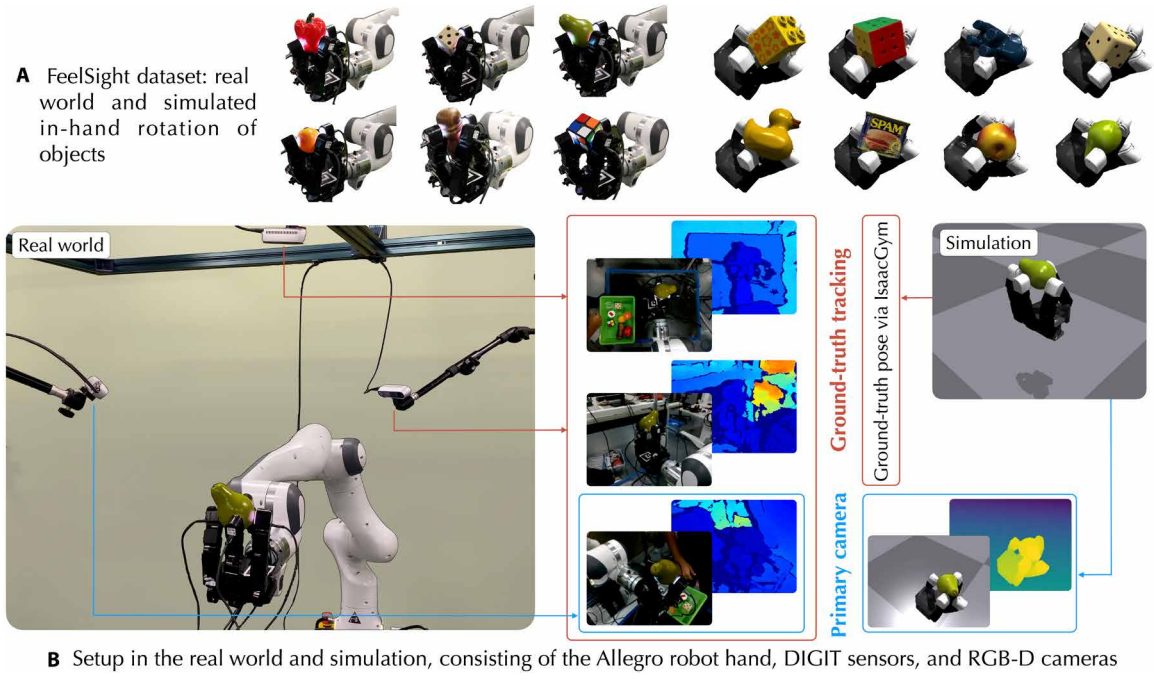


Fig. 7. Setup for real-world and simulation experiments. (**A**) Still frames of the Allegro robot manipulating objects from the FeelSight dataset with our in-hand rotation policy. These visuotactile interactions are captured across the real-world and physics simulation. (**B**) The robot cell was made 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 used the primary camera and DIGIT sensing, and the additional cameras were fused for our ground-truth pose tracking. In simulation, we simulated an identical primary camera in Isaac Gym with simulated touch from the TACTO simulator.

scalar signed distance from the object's closest surface. Online updates were decomposed into alternating steps between refining the weights of the neural SDF θ and optimizing the object pose x_t. Our bespoke object model represents both the pose and object geometry over time. This is further described in the "Object model" section of Materials and Methods.

Given RGB-D vision, RGB touch, and proprioception inputs, our front end returned segmented depth measurements compatible with our back-end optimizer. These modules were pretrained with a large corpus of data. The shape optimizer used the front-end output and optimized for θ at a fixed object pose $\overline{x}_t$ via gradient descent (*29*). Each shape iteration resulted in improved object SDF $F^{\theta}_{\overline{x}_t}$. Last, the pose optimizer built and solved an object pose graph (*81*) for x_t given fixed network weights $\overline{\theta}$. Every pose iteration spatially aligned the evolving object SDF with the current set of front-end output. This is further described in the "Front end" and "Back end: Shape and pose optimizer" sections of Materials and Methods.

Key insights

NeuralFeels is a posed neural field

The object model $F^{\theta}_{x_t}$ is estimated by an alternating optimization of both the neural field weights θ and the object pose x_t. A prior work estimated the pose of a sensor in a trained neural field by "inverting" this optimization—iNeRF (*37*) is a key example of this idea. Other works looked at jointly optimizing the weights of the neural field and pose (*32*, *33*, *105*). In our case, robot kinematics gave us the pose of the touch sensors, and extrinsics gave us the pose of the camera. Thus, we instead flipped this paradigm to estimate the pose of the neural field with respect to known pose sensors.

Touch is vision, albeit local

Another insight is that vision-based touch could be approximated as a perspective camera model in tactile simulators like TACTO (*24*). There were, however, differences that must be accounted for in image formation. First, vision-based tactile sensors imposed their own color and illumination to the scene, which made it hard to get reliable visual cues. Second, a tactile image stream had considerably smaller metric field-of-view and depth range, which was usually in centimeters rather than meters. Third, tactile images had depth discontinuities along all noncontact regions, as opposed to natural images, which only had them along occlusion boundaries. Our method addressed these challenges given that it consistently used depth rather than color for optimization, sampled at different scales (centimeter versus meter) on the basis of the sensing source, and sampled only surface points for touch but both free-space and surface points for vision. More details are in the "Back end: Shape and pose optimizer" section of Materials and Methods.

Object model

In general, a neural SDF (*29*, *31*, *106*) represents 3D surfaces as the zero-level set of a learnable function $F(\mathbf{p}) : \mathbb{R}^3 \rightarrow \mathbb{R}$. The scalar field's sign indicates whether any query point $\mathbf{p}$ in the volume is inside (negative), outside (positive), or on (≈0) the reconstructed surface. $\mathbf{p}$ is first positionally encoded (*107*) into a higher-dimensional space, which helped the network better approximate high-frequency surfaces. This is followed by a multilayer perceptron (MLP) that fit 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 (*108*).

Neural SDFs are more compact than the more popular neural radiance fields (*28*), given that they do not model color and appearance properties. This was sufficient for manipulation because we cared more about estimating geometry than generating novel views. Recently, Instant-NGP (*29*) demonstrated a learnable multiresolution hash table as a positional encoding that greatly accelerates SDF optimization with small MLP backbones. This had been successfully leveraged for real-time SLAM in an indoor scene (*105*). In our work, $F^{\theta}_{x_t}$ represented the neural SDF of the object at a given pose x_t. x_0 was initialized to be at the object ground truth, and θ was randomly initialized. Both shape and pose were estimated via alternating optimization, emulating the paradigm of tracking and mapping that had achieved success in robot vision (*82*).

Front end

The front end, shown in the center column of Fig. 2, extracted depth measurements from raw vision and touch sensing. Depth was available as is in an RGB-D camera, but the challenge was to robustly segment out object depth pixels in occluded interactions. Toward this, we introduced a kinematics-aware segmentation strategy using powerful vision foundation models (*109*). For vision-based touch, estimating depth from images was an open research problem (*22*, *25*, *26*, *66*, *110*). Toward this, we presented a transformer architecture that accurately predicted DIGIT contact patches from input images. Both of the front-end networks were pretrained from a large corpus of data. The output of our front end was a segmented depth image $\widehat{D}^s_t$ for each sensor $s \in \mathcal{S}$.

Segmented visual depth

Robust segmentation of the image stream I^c_t had successfully been demonstrated by image foundation models, like SAM (*109*). Trained with a vision transformer (ViT) in the data-rich natural image domain, SAM generalized to novel scenes for state-of-the-art, zero-shot instance segmentation. For any input RGB image, SAM outputs an embedding that must be queried by user prompts (such as point, binary mask, bounding box, or natural language prompts). At time step t, we fed the model both positive and negative point prompts alongside the mask prediction from time step $t-1$.

Through robot proprioception (refer to Fig. 2), we obtained the four fingertip positions $\mathbf{p}_f$ and computed the centroid $\mathbf{p}_c = \overline{\mathbf{p}}_f$. Given our camera c with known projection operation Π^c, we could obtain any such 3D point $\mathbf{p}$ as a pixel $(u, v) = \Pi^c(\mathbf{p})$ on the image I^c_t. Assuming that the object exists in hand, the centroid pixel $\Pi^c(\mathbf{p}_c)$ served as a useful positive prompt.

In practice, this prompt alone did not suffice—the robot hand frequently appeared in segmentation causing large errors in back-end optimization. To address this, we first rendered the current object model $F^{\theta}_{x_t}$ onto camera c to check whether it occluded any fingertip pixels $\Pi^c(\mathbf{p}_f)$. Each unoccluded fingertip pixel $\Pi^c(\mathbf{p}_f)$ was used as a negative prompt.

In Fig. 8A, we visualized the segmentation on real-world images, alongside the SAM prompts. In our experiments, we used the ViT Large model with 308 million parameters. This achieved a speed of around 4 Hz, but in practice, we could use efficient segmentation models (*91*) for speeds up to 40 Hz. The "Additional implementation details" section of the Supplementary Materials highlights the steps we took for robust visual segmentation.

Tactile transformer

In contrast, vision-based touch images were out of distribution from images SAM was trained on and did not directly provide depth either. The embedded camera perceives an illuminated gel pad, and the contact depth is either obtained via photometric stereo (*16*) or supervised learning (*22*, *25*, *26*, *66*, *110*). Existing touch-to-depth relies on convolution; however, a recent work has shown the benefit of a ViT for dense depth prediction (*111*) in natural images. We trained a tactile transformer for predicting the contact depth from vision-based touch to generalize across multiple real-world DIGIT sensors.

The architecture was trained entirely in tactile simulation, using weights initialized from a pretrained image-to-depth model (*111*). The tactile transformer represented the inverse sensor model $\Omega: I^s_t \mapsto \widehat{D}^s_t$ where $s \in \{d_{\text{index}}, d_{\text{middle}}, d_{\text{ring}}, d_{\text{thumb}}\} \in \mathcal{S}$. This architecture was based on the dense ViT (*111*) and was lightweight (21.7 M parameters) compared with its fully convolutional counterparts (*13*).

Similar to prior works (*13*, *26*), we generated a large corpus of tactile images and paired ground-truth depth maps in the optical touch simulator TACTO (*24*). We collected 10,000 random tactile interactions each on the surfaces of 40 unique YCB objects (*103*). For sim-to-real transfer, we augmented the data with randomization in sensor light-emitting diodes, indentation depth, and pixel noise. In TACTO, image realism was achieved by compositing with template noncontact images from real-world DIGITs. For more details on the training and data, refer to the "Tactile transformer: Data and training" section of the Supplementary Materials.

These augmentations enabled generalized performance across our multifinger platform, where each sensor had differing image characteristics. Our tactile transformer was supervised on mean square depth reconstruction loss against the ground-truth depth maps from simulation. On the basis of the predicted depth maps, the output was thresholded to mask out noncontact regions. We demonstrated an average prediction error of 0.042 mm on a simulated test set, and Fig. 8B shows sim-to-real performance on real-world images.

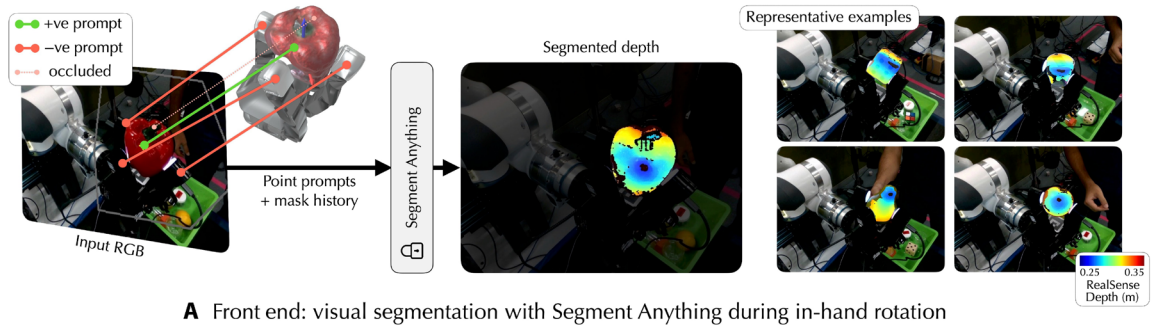
Back end: Shape and pose optimizer

The back end took depth and sensor poses from the front end to build our object model online. This alternated between shape and pose optimization steps using samples from the visuotactile depth stream. Similar to other neural SLAM methods (*31*), the modules maintained a bank of keyframes over time, similar to the strategies of Ortiz *et al.* (*31*) and Sucar *et al.* (*32*), to generate these samples. More details of the back end and keyframing are found in the "Additional implementation details" section of the Supplementary Materials.

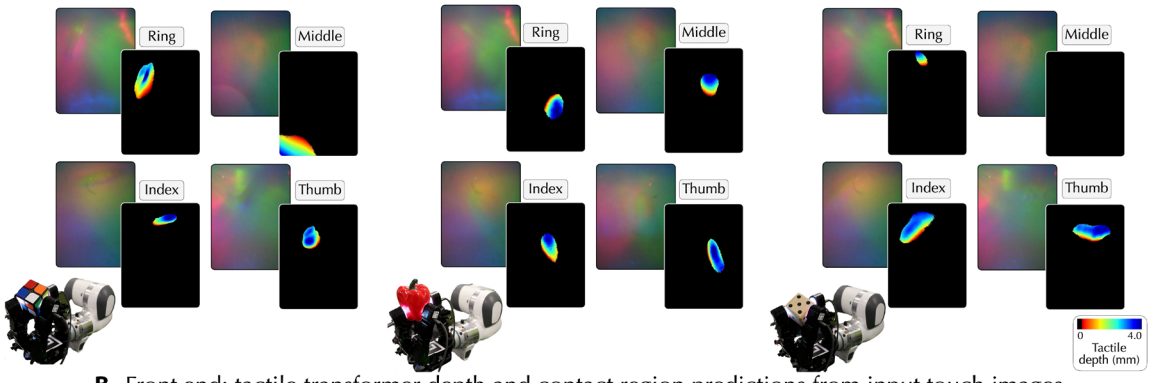
Shape optimizer

For online estimation, it was intractable to optimize $F^{\theta}_{\overline{x}_t}$ using all input frames as in neural radiance fields (*28*). We opted for an online learning approach (*31*, *32*), which built a subset of keyframes $\mathcal{K}$ on the fly to optimize over. The back end would both accept new keyframes on the basis of a criterion and replay old keyframes in the optimization to prevent catastrophic forgetting (*32*). Each iteration of the shape optimizer replayed a batch $k_t \in \mathcal{K}$ of size 10 per sensor to optimize our network. This included the latest two frames and a weighted random sampling of past keyframes based on average rendering loss. The initial visuotactile frame was automatically added as a keyframe

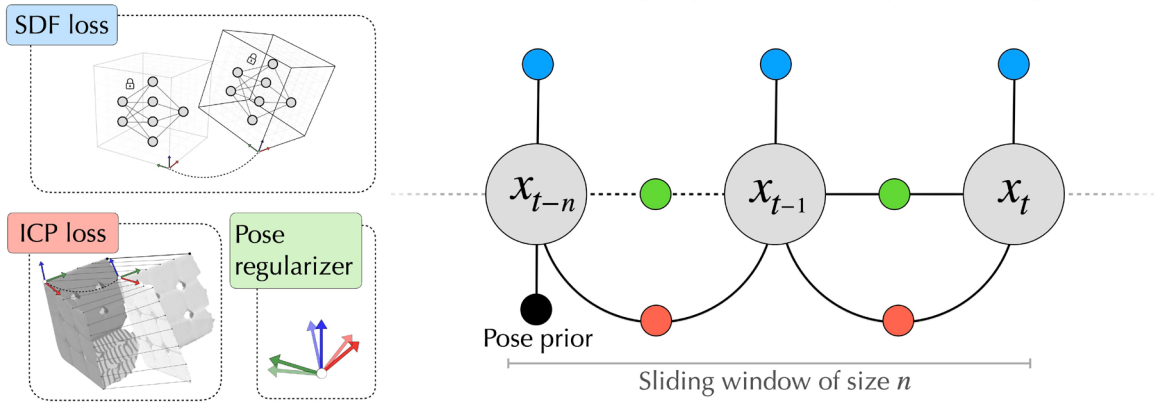
$$\mathcal{K}_0 = \{\, \widehat{D}^s_0 \;\mid\; s \in \mathcal{S} \,\} \tag{1}$$



A Front end: visual segmentation with Segment Anything during in-hand rotation



B Front end: tactile transformer depth and contact region predictions from input touch images



C Back end: pose graph optimization

Fig. 8. Front end and back end. (**A**) Through reasoning about finger occlusion and object pose with respect to the fingers, we could accurately prompt SAM (*109*) for robust output masks. (**B**) Representative examples of the sim-to-real performance of the tactile transformer. Each RGB image was fed through the network to output a predicted depth, along with a contact mask. (**C**) Our sliding window nonlinear least-squares optimizer estimated the object pose x_t from the outputs of the front end. Each object pose x_t was constrained by the SDF loss, frame-to-frame ICP, and pose regularization to ensure that tracking remains stable.

and every subsequent keyframe $\mathcal{K}_t$ was accepted using an information gain metric (*32*). For this, the average rendering loss was computed from the frozen network $F_{x_t}^{\theta}$ using the given keyframe pose and compared against a threshold $d_{\text{thresh}} = 0.01$ m. Last, if we had not added a keyframe for an interval $t_{\max} = 0.2$ s, then we forced one to be added.

Sampling and SDF loss

At each iteration, we sampled coordinates in the neural volume from k_t to optimize the neural weights θ. The first step was to sample a batch of pixels $\mathbf{u}_{kt}$ from k_t—a mix of surface and free-space pixels. The surface pixels directly supervised the SDF zero-level set, and free-space pixels carved out the neural volume. In our implementation, we sampled 50% of the camera pixels in free space, although we only sampled surface pixels for touch. Through each pixel $u \in \mathbf{u}_{kt}$ given their corresponding sensor pose, we projected a ray into the neural volume. Similar to Ortiz *et al.* (*31*), we sampled P_u points per ray, a mix of stratified and surface points.

With these samples, we computed an SDF prediction $\hat{\mathbf{d}}_u$ for each $\hat{D}_t \in k_t$, as the batch distance bound (*31*). For each ray, we split the samples into P_u^{f} and P_u^{tr} on the basis of whether $\hat{\mathbf{d}}_u$ was within the truncation distance $d_{\text{tr}} = 5$ mm from the surface. Our shape loss resembled the truncated SDF loss of Azinović *et al.* (*106*)

$$\mathscr{L}_{\text{shape}} = \mathscr{L}_{\text{f}} + w_{\text{tr}}\mathscr{L}_{\text{tr}}, \text{ with } w_{\text{tr}} = 10 \tag{2}$$

where the free-space and truncated losses were as follows

$$\begin{aligned}\mathscr{L}_{\text{f}} &= \frac{1}{|\mathbf{u}_{k_t}|}\sum_{u\in\mathbf{u}_{k_t}}\frac{1}{|P_u^{\text{f}}|}\left|F_{\bar{x}_t}^{\bar{\theta}}\left(P_u^{\text{f}}\right) - d_{\text{tr}}\right| \text{ and}\\ \mathscr{L}_{\text{tr}} &= \frac{1}{|\mathbf{u}_{k_t}|}\sum_{u\in\mathbf{u}_{k_t}}\frac{1}{|P_u^{\text{tr}}|}\left|F_{\bar{x}_t}^{\bar{\theta}}\left(P_u^{\text{tr}}\right) - \hat{\mathbf{d}}_u\right|\end{aligned} \tag{3}$$

Pose optimizer

Before each shape iteration, we used a pose graph (*75*) to refine the object pose x_t with respect to the frozen neural field $F_{x_t}^{\bar{\theta}}$. We achieved this by inverting the problem to instead optimize for the six DoF poses in a sliding window of size n. At time step t, if we had accumulated N keyframes, then this represented poses $\mathcal{X}_t = \left(x_i\right)_{N-n\le i\le N}$ and measurements $\mathcal{M}_t = \left(\hat{D}_i^s \mid s\in\mathcal{S}\right)_{N-n\le i\le N}$. Similar to pose updates in visual SLAM (*32*, *33*, *37*), the network weights $\bar{\theta}$ were frozen, and we estimated that the $SE(3)$ poses $\mathcal{X}_t$ instead.

We formulated the problem as a nonlinear least-squares optimization with custom measurement factors in Theseus (*81*). Although a prior work used gradient descent (*37*), we instead used a second-order LM solver, which provided faster convergence (*75*). The pose graph, illustrated in Fig. 8C, solved for the following factors

$$\begin{aligned}\hat{\mathcal{X}}_t &= \underset{\mathcal{X}_t}{\text{argmin}}\ \mathscr{L}_{\text{pose}}\left(\mathcal{X}_t \mid \mathcal{M}_t, \bar{\theta}\right) \text{ where}\\ \mathscr{L}_{\text{pose}} &= w_{\text{sdf}}\mathscr{L}_{\text{sdf}} + w_{\text{reg}}\mathscr{L}_{\text{reg}} + w_{\text{icp}}\mathscr{L}_{\text{icp}}\end{aligned} \tag{4}$$

Our SDF loss $\mathscr{L}_{\text{sdf}}$ factor used the previously defined shape loss $\mathscr{L}_{\text{shape}}$, modified such that we sampled only about surface points of each ray. This worked well for both visual and tactile sensing given that 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 sampled surface points over M rays and averaged the SDF loss along each ray. This resulted in an $M \times n$ SDF loss, which we used to update the se(3) lie algebra of $\mathcal{X}_t$. We implemented a custom Jacobian for this cost function, which was up to four times more efficient than PyTorch automatic differentiation.

The pose regularizer $\mathscr{L}_{\text{reg}}$ factor applied a weak regularizer between consecutive keyframe poses in $\mathcal{X}_t$ to ensure that the relative pose updates stayed well behaved. This was important for robustness to noisy front-end depth and incorrect segmentations. We further added an ICP loss $\mathscr{L}_{\text{icp}}$ factor that applied an ICP between the current visuotactile point cloud $\Pi^{-1}(\mathcal{M}_t)$ and previous point cloud $\Pi^{-1}(\mathcal{M}_{t-1})$. This gave us frame-to-frame constraints in addition to the frame-to-model $\mathscr{L}_{\text{sdf}}$.

Statistical analysis

All P values presented in the paper were computed via the paired samples t test (*112*) and are reported as ($P \le \cdot$). Aggregated statistics in Figs. 3 (A and B) and 5C were computed over a combined 70 trials (40 in simulation and 30 in the real world), with each trial run over five different seeds. The boxplots in Fig. 3 (A and B) have whiskers that span 1.0× the interquartile range (IQR), whereas the log-scale plot in Fig. 5C spans 0.25× the IQR. Last, the line plot in Fig. 6B is shaded to represent 1 SD from the mean.

Supplementary Materials

The PDF file includes:

Supplementary Sections
Figs. S1 to S24
Tables S1 to S4
References (*113*–*115*)

Other Supplementary Material for this manuscript includes the following:

Movies S1 to S3

REFERENCES AND NOTES

1. H. Moravec, *Mind Children: The Future of Robot and Human Intelligence* (Harvard Univ. Press, 1988).
2. A. J. Davison, FutureMapping: The computational structure of spatial AI systems. arXiv:1803.11288 [cs.AI] (2018).
3. OpenAI, M. Andrychowicz, B. Baker, M. Chociej, R. Józefowicz, B. M. Grew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. Weng, W. Zaremba, Learning dexterous in-hand manipulation. *Int. J. Robot. Res.* **39**, 3 (2020).
4. Open AI, I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. M. Grew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, J. Schneider, N. Tezak, J. Tworek, P. Welinder, L. Weng, Q. Yuan, W. Zaremba, L. Zhang, Solving Rubik's Cube with a robot hand. arXiv:1910.07113 [cs.LG] (2019).
5. A. Handa, A. Allshire, V. Makoviychuk, A. Petrenko, R. Singh, J. Liu, D. Makoviichuk, K. Van Wyk, A. Zhurkevich, B. Sundaralingam, Y. S. Narang, J.-F. Lafleche, D. Fox, G. State, DeXtreme: Transfer of agile in-hand manipulation from simulation to reality, in *2023 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2023), pp. 5977–5984.
6. H. Qi, B. Yi, S. Suresh, M. Lambeta, Y. Ma, R. Calandra, J. Malik, General in-hand object rotation with vision and touch, in *Proceedings of the 7th Conference on Robot Learning (CoRL)* (ML Research Press, 2023), pp. 1722–1732.
7. Z.-H. Yin, B. Huang, Y. Qin, Q. Chen, X. Wang, Rotating without seeing: Towards in-hand dexterity through touch, in *Proceedings of Robotics: Science and Systems* (RSS Foundation, 2023).
8. I. Guzey, B. Evans, S. Chintala, L. Pinto, Dexterity from touch: Self-supervised pre-training of tactile representations with robotic play, in *Proceedings of the 7th Conference on Robot Learning (CoRL)* (ML Research Press, 2023), pp. 3142–3166.
9. I. Guzey, Y. Dai, B. Evans, S. Chintala, L. Pinto, See to touch: Learning tactile dexterity through visual incentives, in *2024 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2024), pp. 13825–13832.
10. T. Chen, M. Tippur, S. Wu, V. Kumar, E. Adelson, P. Agrawal, Visual dexterity: In-hand reorientation of novel and complex object shapes. *Sci. Robot.* **8**, eadc9244 (2023).
11. H. Qi, A. Kumar, R. Calandra, Y. Ma, J. Malik, In-hand object rotation via rapid motor adaptation, in *Proceedings of the 6th Conference on Robot Learning (CoRL)* (ML Research Press, 2022), pp. 1722–1732.
12. Y. She, S. Wang, S. Dong, N. Sunil, A. Rodriguez, E. Adelson, Cable manipulation with a tactile-reactive gripper. *Int. J. Robot. Res.* **40**, 1385–1401 (2021).
13. S. Suresh, Z. Si, S. Anderson, M. Kaess, M. Mukadam, MidasTouch: Monte-Carlo inference over distributions across sliding touch, in *Proceedings of the 6th Conference on Robot Learning (CoRL)* (ML Research Press, 2022), pp. 319–331.
14. H. B. Helbig, M. O. Ernst, Optimal integration of shape information from vision and touch. *Exp. Brain Res.* **179**, 595–606 (2007).
15. Z. Kappassov, J.-A. Corrales, V. Perdereau, Tactile sensing in dexterous robot hands. *Robot. Auton. Syst.* **74**, 195–220 (2015).
16. W. Yuan, S. Dong, E. H. Adelson, GelSight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors* **17**, 2762 (2017).
17. E. Donlon, S. Dong, M. Liu, J. Li, E. Adelson, A. Rodriguez, GelSlim: A high-resolution, compact, robust, and calibrated tactile-sensing finger, in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2018), pp. 1927–1934.
18. B. Ward-Cherrier, N. Pestell, L. Cramphorn, B. Winstone, M. E. Giannaccini, J. Rossiter, N. F. Lepora, The TacTip family: Soft optical tactile sensors with 3D-printed biomimetic morphologies. *Soft Robot.* **5**, 216–227 (2018).
19. A. Alspach, K. Hashimoto, N. Kuppuswamy, R. Tedrake, Soft-bubble: A highly compliant dense geometry tactile sensor for robot manipulation, in *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)* (IEEE, 2019), pp. 597–604.
20. M. Lambeta, P. W. Chou, S. Tian, B. Yang, B. Maloon, V. R. Most, D. Stroud, R. Santos, A. Byagowi, G. Kammerer, D. Jayaraman, R. Calandra, DIGIT: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation. *IEEE Robot. Autom. Lett.* **5**, 3838–3845 (2020).

21. A. Padmanabha, F. Ebert, S. Tian, R. Calandra, C. Finn, S. Levine, OmniTact: A multi-directional high-resolution touch sensor, in *2020 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2020), pp. 618–624.
22. S. Wang, Y. She, B. Romero, E. Adelson, GelSight wedge: Measuring high-resolution 3D contact geometry with a compact robot finger, in *2021 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2021), pp. 6468–6475.
23. W. K. Do, M. Kennedy, Densetact: Optical tactile sensor for dense shape reconstruction, in *2022 International Conference on Robotics and Automation (ICRA)* (IEEE, 2022), pp. 6188–6194.
24. S. Wang, M. M. Lambeta, P.-W. Chou, R. Calandra, TACTO: A fast, flexible, and open-source simulator for high-resolution vision-based tactile sensors. *IEEE Robot. Autom. Lett.* **7**, 3930–3937 (2022).
25. P. Sodhi, M. Kaess, M. Mukadam, S. Anderson, PatchGraph: In-hand tactile tracking with learned surface normals, in *2022 International Conference on Robotics and Automation (ICRA)* (IEEE, 2022), pp. 2164–2170.
26. S. Suresh, Z. Si, J. G. Mangelson, W. Yuan, M. Kaess, ShapeMap 3-D: Efficient shape mapping through dense touch and vision, in *2022 International Conference on Robotics and Automation (ICRA)* (IEEE, 2022), pp. 7073–7080.
27. Y. Xie, T. Takikawa, S. Saito, O. Litany, S. Yan, N. Khan, F. Tombari, J. Tompkin, V. sitzmann, S. Sridhar, Neural fields in visual computing and beyond. *Comput. Graph. Forum* **41**, 641–676 (2022).
28. B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, R. Ng, NeRF: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM* **65**, 99–106 (2021).
29. T. Müller, A. Evans, C. Schied, A. Keller, Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.* **41**, 1–15 (2022).
30. Z. Li, T. Müller, A. Evans, R. H. Taylor, M. Unberath, M.-Y. Liu, C.-H. Lin, Neuralangelo: High-fidelity neural surface reconstruction, in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2023), pp. 8456–8465.
31. J. Ortiz, A. Clegg, J. Dong, E. Sucar, D. Novotny, M. Zollhoefer, M. Mukadam, iSDF: Real-time neural signed distance fields for robot perception, in *Proceedings of Robotics: Science and Systems* (RSS Foundation, 2022).
32. E. Sucar, S. Liu, J. Ortiz, A. J. Davison, iMAP: Implicit mapping and positioning in real- time, in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (IEEE, 2021), pp. 6229–6238.
33. Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M. R. Oswald, M. Pollefeys, NICE-SLAM: Neural implicit scalable encoding for SLAM, in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2022), pp. 12786–12796.
34. B. Wen, J. Tremblay, V. Blukis, S. Tyree, T. Muller, A. Evans, D. Fox, J. Kautz, S. Birchfield, BundleSDF: Neural 6-DoF tracking and 3D reconstruction of unknown objects, in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2023), pp. 606–617.
35. A. Yu, V. Ye, M. Tancik, A. Kanazawa, PixelNeRF: Neural radiance fields from one or few images, in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2021), pp. 4578–4587.
36. J. J. Park, P. Florence, J. Straub, R. Newcombe, S. Lovegrove, DeepSDF: Learning continuous signed distance functions for shape representation, in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2019), pp. 165–174.
37. L. Yen-Chen, P. Florence, J. T. Barron, A. Rodriguez, P. Isola, T.-Y. 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.
38. P. Grote, J. Ortiz-Haro, M. Toussaint, O. S. Oguz, Neural field representations of articulated objects for robotic manipulation planning. arXiv:2309.07620 [cs.RO] (2023).
39. S. Le Cleac'h, H.-X. Yu, M. Guo, T. A. Howell, R. Gao, J. Wu, Z. Manchester, M. Schwager, Differentiable physics simulation of dynamics-augmented neural objects. *IEEE Robot. Autom. Lett.* **8**, 2780–2787 (2023).
40. D. Driess, I. Schubert, P. Florence, Y. Li, M. Toussaint, Reinforcement learning with neural radiance fields. *Adv. Neural Inf. Process. Syst.* **35**, 16931 (2022).
41. Y. Wi, A. Zeng, P. Florence, N. Fazeli, VIRDO++: Real-world, visuo-tactile dynamics and perception of deformable objects, in *Proceedings of the 6th Conference on Robot Learning* (ML Research Press, 2023), pp. 1806–1816.
42. Y. Li, S. Li, V. Sitzmann, P. Agrawal, A. Torralba, 3D neural scene representations for visuomotor control, in *Proceedings of the 5th Conference on Robot Learning* (ML Research Press, 2022), pp. 112–123.
43. S. Zhong, A. Albini, O. P. Jones, P. Maiolino, I. Posner, Touching a NeRF: Leveraging neural radiance fields for tactile sensory data generation, in *Proceedings of the 6th Conference on Robot Learning* (ML Research Press, 2022), pp. 1–11.
44. J. Ichnowski, Y. Avigal, J. Kerr, K. Goldberg, Dex-NeRF: Using a neural radiance field to grasp transparent objects, in *Proceedings of the 5th Conference on Robot Learning* (ML Research Press, 2022), pp. 526–536.
45. J. Kerr, L. Fu, H. Huang, Y. Avigal, M. Tancik, J. Ichnowski, A. Kanazawa, K. Goldberg, Evo-NeRF: Evolving NeRF for sequential robot grasping of transparent objects, in *Proceedings of the 6th Conference on Robot Learning* (ML Research Press, 2022), pp. 353–367.
46. M. Moll, M. 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.
47. J. Ilonen, J. Bohg, V. Kyrki, Fusing visual and tactile sensing for 3-D object reconstruction while grasping, in *2013 IEEE International Conference on Robotics and Automation* (IEEE, 2013), pp. 3547–3554.
48. S. Wang, J. Wu, X. Sun, W. Yuan, W. T. Freeman, J. B. Tenenbaum, E. H. Adelson, 3D shape perception from monocular vision, touch, and shape priors, in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2018), pp. 1606–1613.
49. E. J. Smith, R. Calandra, A. Romero, G. Gkioxari, D. Meger, J. Malik, M. Drozdzal, 3D shape reconstruction from vision and touch, in *Proceedings of the 34th International Conference on Neural Information Processing Systems,* H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, H. Lin, Eds. (Curran Associates Inc., 2020), pp. 14193–14206.
50. W. Xu, Z. Yu, H. Xue, R. Ye, S. Yao, C. Lu, Visual-tactile sensing for in-hand object reconstruction, in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2023), pp. 8803–8812.
51. Y. Chen, A. E. Tekden, M. P. Deisenroth, Y. Bekiroglu, Sliding touch-based exploration for modeling unknown object shape with multi-fingered hands, in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2023), pp. 8943–8950.
52. M. Comi, Y. Lin, A. Church, A. Tonioni, L. Aitchison, N. F. Lepora, TouchSDF: A DeepSDF approach for 3D shape reconstruction using vision-based tactile sensing. *IEEE Robot. Automat. Lett.* **9**, 5719–5726 (2024).
53. K.-T. Yu, A. Rodriguez, Realtime state estimation with tactile and visual sensing: Application to planar manipulation, in *2018 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2018), pp. 7778–7785.
54. A. S. Lambert, M. Mukadam, B. Sundaralingam, N. Ratliff, B. Boots, D. Fox, Joint inference of kinematic and force trajectories with visuo-tactile sensing, in *2019 International Conference on Robotics and Automation (ICRA)* (IEEE, 2019), pp. 3165–3171.
55. P. Sodhi, M. Kaess, M. Mukadam, S. Anderson, Learning tactile models for factor graph-based estimation, in *2021 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2021), pp. 13686–13692.
56. A. Petrovskaya, O. Khatib, Global localization of objects via touch. *IEEE Trans. Robot.* **27**, 569–585 (2011).
57. G. M. Caddeo, N. A. Piga, F. Bottarel, L. Natale, Collision-aware in-hand 6d object pose estimation using multiple vision-based tactile sensors, in *2023 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2023), pp. 719–725.
58. K.-T. Yu, J. Leonard, A. Rodriguez, Shape and pose recovery from planar pushing, in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2015), pp. 1208–1215.
59. S. Suresh, M. Bauza, K.-T. Yu, J. G. Mangelson, A. Rodriguez, M. Kaess, Tactile SLAM: Real-time inference of shape and pose from planar pushing, in *2021 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2021), pp. 11322–11328.
60. C. Strub, F. Wörgötter, H. Ritter, Y. 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.
61. C. Pan, M. Lepert, S. Yuan, R. Antonova, J. Bohg, In-hand manipulation of unknown objects with tactile sensing for insertion, in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2023), pp. 8765–8771.
62. J. Zhao, M. Bauza, E. H. Adelson, FingerSLAM: Closed-loop unknown object localization and reconstruction from visuo-tactile feedback, in *2023 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2023), pp. 8033–8039.
63. Wonik Robotics, Allegro Hand, https://allegrohand.com/.
64. J. Tremblay, B. Wen, V. Blukis, B. Sundaralingam, S. Tyree, S. Birchfield, Diff-DOPE: Differentiable deep object pose estimation. arXiv:2310.00463 [cs.CV] (2023).
65. Y. Xiang, T. Schmidt, V. Narayanan, D. Fox, PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes, in *Proceedings of Robotics: Science and Systems* (RSS Foundation, 2018).
66. M. Bauza, O. Canal, A. Rodriguez, Tactile mapping and localization from high-resolution tactile imprints, in *2019 International Conference on Robotics and Automation (ICRA)* (IEEE, 2019), pp. 3811–3817.
67. J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, S. Birchfield, Deep object pose estimation for semantic robotic grasping of household objects, in *Proceedings of the 2nd Conference on Robot Learning* (ML Research Press, 2018), pp. 306–316.
68. A. Knapitsch, J. Park, Q.-Y. Zhou, V. Koltun, Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Trans. Graph.* **36**, 1–13 (2017).
69. M. Tatarchenko, S. R. Richter, R. Ranftl, Z. Li, V. Koltun, T. Brox, What do single-view 3D reconstruction networks learn?, in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2019), pp. 3400–3409.

70. V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, G. State, Isaac Gym: High performance GPU-based physics simulation for robot learning. arXiv:2108.10470 [cs.RO] (2021).
71. Y. Labbé, L. Manuelli, A. Mousavian, S. Tyree, S. Birchfield, J. Tremblay, J. Carpentier, M. Aubry, D. Fox, J. Sivic, Megapose: 6D pose estimation of novel objects via render & compare, in *Proceedings of the 6th Conference on Robot Learning* (ML Research Press, 2023), pp. 715–725.
72. J. Sun, Z. Shen, Y. Wang, H. Bao, X. Zhou, LoFTR: Detector-free local feature matching with transformers, in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), pp. 8922–8931.
73. M. Bauza, A. Bronars, A. Rodriguez, Tac2Pose: Tactile object pose estimation from the first touch. *Int. J. Robot. Res.* **42**, 1185-1209 (2023).
74. P.-E. Sarlin, D. DeTone, T. Malisiewicz, A. Rabinovich, Superglue: Learning feature matching with graph neural networks, in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2020), pp. 4938–4947.
75. F. Dellaert, M. Kaess, Factor graphs for robot perception. *Found. Trends Robot.* **6**, 1–139 (2017).
76. W. E. Lorensen, H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm" in *Seminal Graphics: Pioneering Efforts That Shaped the Field*, R. Wolfe, Ed. (Association for Computing Machinery, 1998), pp. 347–353.
77. M. Bauza, A. Bronars, Y. Hou, I. Taylor, N. Chavan-Dafle, A. Rodriguez, SimPLE, a visuotactile method learned in simulation to precisely pick, localize, regrasp, and place objects. *Sci. Robot.* **9**, eadi8808 (2024).
78. L. Keselman, K. Shih, M. Hebert, A. Steinfeld, Optimizing algorithms from pairwise user preferences, in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2023), pp. 4161–4167.
79. A. Handa, T. Whelan, J. McDonald, A. 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.
80. E. J. Smith, D. Meger, L. Pineda, R. Calandra, J. Malik, A. Romero-Soriano, M. Drozdzal, Active 3D shape reconstruction from vision and touch, in *Proceedings of the 35th International Conference on Neural Information Processing Systems* (Curran Associates Inc., 2024), pp. 16064–16078.
81. B. Amos, S. Anderson, R. T. Q. Chen, D. DeTone, J. Dong, T. Fan, M. Monge, M. Mukadam, J. Ortiz, L. Pineda, P. Sodhi, S. Venkataraman, A. Wang, Theseus: A library for differentiable nonlinear optimization, in *Proceedings of the 36th Conference on Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, A. Oh, Eds. (Curran Associates Inc., 2022), pp. 3801–3818.
82. C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, J. J. Leonard, Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Trans. Robot.* **32**, 1309–1332 (2016).
83. J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, P. P. Srinivasan, Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields, in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (IEEE, 2021), pp. 5855–5864.
84. B. Kerbl, G. Kopanas, T. Leimküehler, G. Drettakis, 3D Gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.* **42**, 1–14 (2023).
85. J. Kerr, H. Huang, A. Wilcox, R. I. Hoque, J. Ichnowski, R. Calandra, K. Goldberg, Self-supervised visuo-tactile pretraining to locate and follow garment features, in *Proceedings of Robotics: Science and Systems* (RSS Foundation, 2023).
86. D. Driess, J.-S. Ha, M. Toussaint, R. Tedrake, Learning models as functionals of signed- distance fields for manipulation planning, in *Proceedings of the 5th Conference on Robot Learning* (ML Research Press, 2022), pp. 245–255.
87. C. Higuera, B. Boots, M. Mukadam, Learning to read braille: Bridging the tactile reality gap with diffusion models. arXiv:2304.01182 [cs.RO] (2023).
88. Z. Si, G. Zhang, Q. Ben, B. Romero, Z. Xian, C. Liu, C. Gan, DIFFTACTILE: A physics-based differentiable tactile simulator for contact-rich robotic manipulation, *The Twelfth International Conference on Learning Representations* (ICLR, 2024).
89. E. Dexheimer, A. J. Davison, Learning a depth covariance function, in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2023), pp. 13122–13131.
90. J. A. Fishel, G. E. Loeb, Sensing tactile microvibrations with the BioTac—Comparison with human sensitivity, in *2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)* (IEEE, 2012), pp. 1122–1127.
91. C. Zhang, D. Han, Y. Qiao, J. U. Kim, S.-H. Bae, S. Lee, C. S. Hong, Faster Segment Anything: Towards lightweight SAM for mobile applications. arXiv:2306.14289 [cs.CV] (2023).
92. M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, F. Dellaert, iSAM2: Incremental smoothing and mapping using the Bayes tree. *Int. J. Robot. Res.* **31**, 216–235 (2012).
93. D. DeTone, T. Malisiewicz, A. Rabinovich, Superpoint: Self-supervised interest point detection and description, in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (IEEE, 2018), pp. 224–236.
94. C.-Y. Wu, J. Johnson, J. Malik, C. Feichtenhofer, G. Gkioxari, Multiview compressive coding for 3D reconstruction, in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2023), pp. 9065–9075.
95. Y. Hong, K. Zhang, J. Gu, S. Bi, Y. Zhou, D. Liu, F. Liu, K. Sunkavalli, T. Bui, H. Tan, LRM: Large reconstruction model for single image to 3D. arXiv:2311.04400 [cs.CV] (2023).
96. B. Sundaralingam, T. Hermans, In-hand object-dynamics inference using tactile fingertips. *IEEE Trans. Robot.* **37**, 1115–1126 (2021).
97. S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, M. J. Marín-Jiménez, Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognit.* **47**, 2280–2292 (2014).
98. P. Furgale, J. Rehder, R. 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.
99. T. Hodan, F. Michel, E. Brachmann, W. Kehl, A. G. Buch, D. Kraft, B. Drost, J. Vidal, S. Ihrke, X. Zabulis, C. Sahin, F. Manhardt, F. Tombari, T.-K. Kim, J. Matas, C. Rother, BOP: Benchmark for 6D object pose estimation, in *Proceedings of the European Conference on Computer Vision (ECCV)* (Springer Nature, 2018), pp. 19–34.
100. A. Geiger, P. Lenz, C. Stiller, R. Urtasun, Vision meets robotics: The KITTI dataset. *Int. J. Robot. Res.* **32**, 1231–1237 (2013).
101. S. James, Z. Ma, D. R. Arrojo, A. J. Davison, RLbench: The robot learning benchmark and learning environment. *IEEE Robot. Autom. Lett.* **5**, 3019–3026 (2020).
102. Revopoint, Revopoint POP 3 3D Scanner, https://revopoint3d.com/.
103. B. Calli, A. Singh, J. Bruce, A. Walsman, K. Konolige, S. Srinivasa, P. Abbeel, A. M. Dollar, Yale-CMU-Berkeley dataset for robotic manipulation research. *Int. J. Robot. Res.* **36**, 261–268 (2017).
104. S. Brahmbhatt, A. Handa, J. Hays, D. Fox, ContactGrasp: Functional multi-finger grasp synthesis from contact, in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2019), pp. 2386–2393.
105. A. Rosinol, J. J. Leonard, L. Carlone, NeRF-SLAM: Real-time dense monocular SLAM with neural radiance fields, in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2023), pp. 3437–3444.
106. D. Azinović, R. Martin-Brualla, D. B. Goldman, M. Nießner, J. Thies, Neural RGB-D surface reconstruction, in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2022), pp. 6290–6301.
107. M. Tancik, P. P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. T. Barron, R. Ng, Fourier features let networks learn high frequency functions in low dimensional domains, in *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds. (Curran Associates, 2020), pp. 7537–7547.
108. J. L. Schonberger, J.-M. Frahm, Structure-from-Motion revisited, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2016), pp. 4104–4113.
109. A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, R. Girshick, Segment Anything, in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)* (IEEE, 2023), pp. 4015–4026.
110. R. Ambrus, V. Guizilini, N. Kuppuswamy, A. Beaulieu, A. Gaidon, A. Alspach, Monocular depth estimation for soft visuotactile sensors, in *2021 IEEE 4th International Conference on Soft Robotics (RoboSoft)* (IEEE, 2021), pp. 643–649.
111. R. Ranftl, A. Bochkovskiy, V. Koltun, Vision transformers for dense prediction, in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (IEEE, 2021), pp. 12179–12188.
112. A. Ross, V. L. Willson, A. Ross, V. L. Willson, "Paired samples T-test" in *Basic and Advanced Statistical Tests: Writing Results Sections and Creating Tables and Figures* (SensePublishers, 2017), pp. 17–19.
113. A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, in *The Ninth International Conference on Learning Representations* (ICLR, 2021).
114. D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, in *The Third International Conference on Learning Representations* (ICLR, 2015).
115. J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms. arXiv:1707.06347 [cs.LG] (2017).

Acknowledgments: We thank D. Batra, T. Gervet, and A. Rai for feedback on the writing and W. Dong, T. Hellebrekers, C. Higuera, P. Lancaster, F. Meier, A. Rodriguez, A. Sharma, and J. Yin for helpful discussions on the research. **Funding:** S.S. and H.Q. acknowledge funding from Meta, and their work was partially conducted while at FAIR, Meta. S.S. was further partially supported by NSF grant IIS-2008279 while at CMU. R.C. acknowledges support from the German Research Foundation (DFG, Deutsche Forschungsgemeinschaft) as part of Germany's Excellence Strategy, EXC 2050/1, Project ID 390696704, Cluster of Excellence "Centre for Tactile Internet with Human-in-the-Loop" (CeTI) of Technische Universität Dresden and from Bundesministerium für Bildung und Forschung (BMBF) and German Academic Exchange Service (DAAD), project 57616814 (School of Embedded and Composite AI). **Author contributions:** S.S. developed and implemented the core approach including tactile

transformer, visual depth segmentation, neural SDF reconstruction, and pose-graph optimization; performed full-stack tuning; worked on Allegro and DIGIT integration, TACTO and Isaac Gym integration, camera and robot calibration, data collection, ground-truth object scans, and live visualizations; conducted evaluations; made visuals; and wrote the paper. H.Q. designed and implemented in-hand object rotation policies and sim-to-real policy transfer; helped with Allegro and DIGIT integration, TACTO and Isaac Gym integration, and data collection; performed code reviews and bug fixes; and helped edit the paper. T.W. coordinated hardware and software systems integration; performed profiling of software stack; helped with Allegro and DIGIT integration, camera and robot calibration, and ground-truth object scans; and advised on evaluations. T.F. designed and implemented forward kinematics; helped implement visual depth segmentation, pose-graph cost functions and optimization, and software systems integration; and advised on evaluations. L.P. implemented the workflow for cluster deployment, streamlined development workflow, helped with modules that use Theseus, performed code reviews and bug fixes, and advised on evaluations. M.L. helped with Allegro and DIGIT integration, TACTO and Isaac Gym integration, and hardware systems integrations. J.M. advised on the project and gave feedback on the approach, evaluations, and the paper. Mr.K. advised on the project; managed and supported researchers; and gave feedback on the approach, evaluations, and the paper. R.C. advised on the project; helped with Allegro and DIGIT integration and TACTO and Isaac Gym integration; and gave feedback on the approach, evaluations, and the paper. Mi.K. advised on the project; helped design pose-graph optimization; and gave feedback on the approach, evaluations, and the paper. J.O. advised on the project; codeveloped the core approach; implemented volumetric ray sampling, SDF cost function, and 2D live visualizations; helped implement the workflow for cluster deployment; streamlined the development workflow; performed code reviews and bug fixes; gave feedback on evaluations; designed visuals; and edited the paper. M.M. set the vision and research direction; steered and aligned the team; provided guidance on all aspects of the project, including the core approach, systems, and evaluations; designed visuals; and edited the paper. **Competing interests:** The authors declare that they have no competing interests. **Data and materials availability:** All data to validate the paper are available in the main body and the Supplementary Materials. For multimedia, code, and data, we refer the readers to the project webpage https://suddhu.github.io/neural-feels. The code and data may also be accessed through https://doi.org/10.5061/dryad.b2rbnzsqr.

Submitted 15 December 2023
Accepted 15 October 2024
Published 13 November 2024
10.1126/scirobotics.adl0628

Science Robotics

NeuralFeels with neural fields: Visuotactile perception for in-hand manipulation

Sudharshan Suresh, Haozhi Qi, Tingfan Wu, Taosha Fan, Luis Pineda, Mike Lambeta, Jitendra Malik, Mrinal Kalakrishnan, Roberto Calandra, Michael Kaess, Joseph Ortiz, and Mustafa Mukadam

Sci. Robot. **9** (96), eadl0628. DOI: 10.1126/scirobotics.adl0628

Editor's summary

In-hand perception using neural fields to endow robots with human levels of perception and dexterity is an ongoing problem in robotics. To estimate an object's shape during manipulation, Suresh *et al.* trained a neural field to represent the spatial information of an object using the information gathered from vision and touch. A multifinger robotic hand with vision-based touch sensors rotated an object to gather tactile signals, which were combined with visual data from a stationary camera and input into an online neural field. The neural field used simultaneous localization and mapping (SLAM) to output the pose and geometry of the object. The pipeline, called NeuralFeels, could achieve reconstruction of novel objects with high precision. —Melisa Yashinski

Science Robotics (ISSN 2470-9476) is published by the American Association for the Advancement of Science. 1200 New York Avenue NW, Washington, DC 20005. The title *Science Robotics* is a registered trademark of AAAS.