NormalFlow: Fast, Robust, and Accurate Contact-based Object 6DoF Pose Tracking with Vision-based Tactile Sensors

Hung-Jui Huang[1], Michael Kaess[1], and Wenzhen Yuan[2]

***Abstract*— Tactile sensing is crucial for robots aiming to achieve human-level dexterity. In dexterous manipulation, it plays a critical role in monitoring contact modes and estimating an object's 6DoF pose, both of which are necessary for precise control in multi-fingered hands. In this work, we present NormalFlow, a fast, robust, and real-time tactile-based algorithm that jointly tracks object 6DoF pose and monitors contact. Leveraging the precise surface normal estimation of vision-based tactile sensors, NormalFlow determines object movements by minimizing discrepancies between the tactile-derived surface normals. Our results show that NormalFlow consistently outperforms competitive baselines and can track low-texture objects like flat table surfaces and ping pong balls. Additionally, we present state-of-the-art tactile-based 3D reconstruction results, showcasing the high accuracy of NormalFlow. We believe NormalFlow unlocks new possibilities for high-precision perception and manipulation tasks that involve interacting with objects and tools using hands. The video demo, code, and dataset are available on our website: https://joehjhuang.github.io/normalflow.**

I. Introduction

The skill to interact with and manipulate objects is fundamental for diverse robotics applications. As robots take on more dexterous tasks, accurate in-hand object tracking and contact monitoring becomes critical. These capabilities enable fine-grained control, which is essential for a variety of tasks, including in-hand reorientation and precise tool use, such as handwriting or manipulating chopsticks. Although vision-based systems are widely used for object tracking, they often suffer from occlusion during manipulation. OpenAI's dexterous manipulation system [1] highlights this challenge, as it uses 19 cameras from every angle just to track a block's rotation in hand. Fortunately, vision-based tactile sensors like GelSight [2] offers a way to accurately track objects and monitor contacts without occlusion issues.

In this work, we aim to accurately track objects during contact using vision-based tactile sensors without needing object 3D models. While this capability is the cornerstone of many downstream tasks such as manipulation, dexterous manipulation, and tactile-based 3D reconstruction, the field has not adequately addressed it. Prior works [3], [4], [5], [6] handle object tracking by converting tactile images into point clouds and applying registration methods like ICP [7], but these often perform poorly due to the noise and distortion in tactile-derived point clouds. In this work, we introduce NormalFlow, a state-of-the-art tactile tracking algorithm that outperforms point cloud registration methods in both accuracy and speed. By directly minimizing discrepancies between surface normal maps—rather than relying on point clouds—NormalFlow achieves fast, robust, and accurate 6DoF pose estimation without object models, even on low-texture surfaces like flat table surfaces and ping pong balls. It achieves a mean translation error of 0.29mm (over a total movement of 3.4mm), rotation error of 1.9° (over a total movement of 37.4°), running at 70Hz on CPU. We also demonstrate its application in tactile-based 3D reconstruction, producing high-quality geometry. We believe NormalFlow opens new avenues for higher precision tactile-dependent perception and control, with broad applications including dexterous manipulation.

[1]Hung-Jui Huang and Michael Kaess are with Carnegie Mellon University, Pittsburgh, PA, USA {hungjuih, kaess}@andrew.cmu.edu
[2]Wenzhen Yuan is with University of Illinois Urbana-Champaign, Champaign, IL, USA yuanwz@illinois.edu

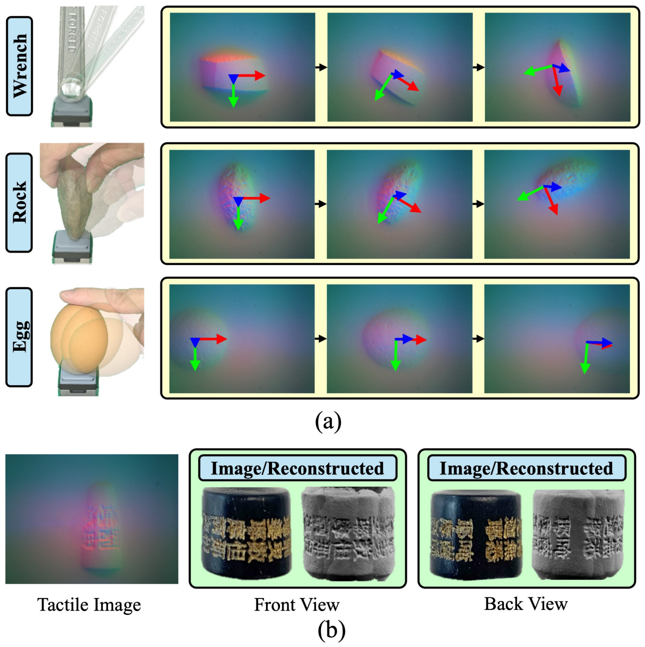

Fig. 1: NormalFlow performs fast, accurate, and robust 6DoF object tracking based on only touch sensing. (a) Accurate tracking of a wide variety of objects, including a wrench, a rock, and even low-texture object like an egg. (b) Applying NormalFlow to tactile-based 3D reconstruction of a 12mm wide bead highlights NormalFlow's high accuracy.

II. Method

NormalFlow tracks object motion by directly minimizing differences between surface normal maps extracted from

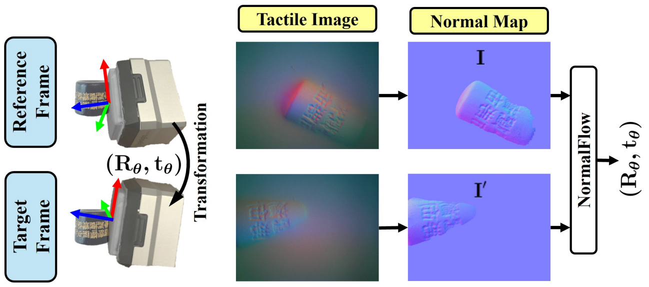

Fig. 2: Given two tactile images before and after object movement, we derive the surface normal maps. NormalFlow determines the object transformations by minimizing discrepancies between the surface normal maps.

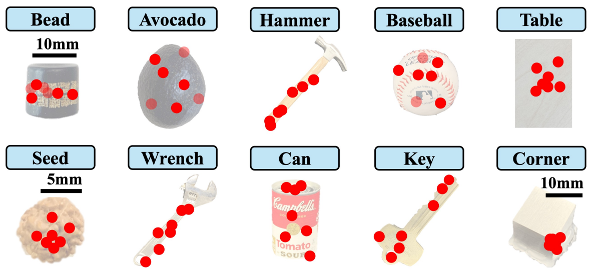

Fig. 3: Initial contact locations (manually labeled) for the seven trials per object in the tracking experiment.

tactile images. We adapt the approach from [8] to estimate these maps. Let $\mathbf{I}$ and $\mathbf{I}'$ denote the surface normal maps of a reference and a target sensor frame, respectively, where each map is a function $\mathbb{R}^2 \mapsto \mathbb{R}^3$ from pixel coordinates to surface normals (Fig. 2). Our goal is to estimate the 6DoF transformation from the reference frame to the target frame $(\mathbf{R}_{\boldsymbol{\theta}}, \mathbf{t}_{\boldsymbol{\theta}}) \in SE(3)$, parameterized as $\boldsymbol{\theta} = (x, y, z, \theta_x, \theta_y, \theta_z) \in \mathbb{R}^6$. The NormalFlow algorithm minimizes the difference between two surface maps: the transformed reference map $\mathbf{I}$ when applying the 3D transformation to the object, and the target map $\mathbf{I}'$. This 3D transformation affects the reference map $\mathbf{I}$ by rotating the normal directions and re-mapping the pixels. Therefore, the minimization objective of NormalFlow is:

$$\sum_{(u,v)\in\overline{C}} \left[\mathbf{I}'(\mathbf{W}(u,v;\boldsymbol{\theta})) - \mathbf{R}_{\boldsymbol{\theta}}\mathbf{I}(u,v)\right]^2 \tag{1}$$

where (u, v) is the pixel coordinates and $\overline{C}$ is the shared contact region. The re-mapping function $\mathbf{W}(u, v; \boldsymbol{\theta})$ maps pixel coordinates from the reference frame to the target frame by transforming the 3D surface coordinate of the pixels and projects it to 2D:

$$\mathbf{W}(u,v;\boldsymbol{\theta}) = \mathbf{P}\big(\mathbf{R}_{\boldsymbol{\theta}} \cdot \mathbf{q}(u,v) + \mathbf{t}_{\boldsymbol{\theta}}\big) \tag{2}$$

where $\mathbf{q}(u,v) = \begin{bmatrix} u & v & z(u,v)\end{bmatrix}^\intercal$ is the 3D surface coordinate corresponding to the pixel at (u, v) in the reference frame and $\mathbf{P} = \left(\begin{smallmatrix}1&0&0\\0&1&0\end{smallmatrix}\right)$ is the projection matrix. Inspired by the Lucas-Kanade optical flow method [9] [10], NormalFlow employs the Gauss-Newton optimization to minimize Eq. (1) iteratively. Linearizing Eq. (1) at the current estimate of $\boldsymbol{\theta}$ results in:

$$\sum_{(u,v)\in\overline{C}} \Big[\big(\mathbf{I}'(\mathbf{W}) - \mathbf{R}_{\boldsymbol{\theta}}\mathbf{I}\big) + \big(\nabla\mathbf{I}'\frac{\partial\mathbf{W}}{\partial\boldsymbol{\theta}} - \frac{\partial(\mathbf{R}_{\boldsymbol{\theta}}\mathbf{I})}{\partial\boldsymbol{\theta}}\big)\boldsymbol{\Delta\theta}\Big]^2 \tag{3}$$

We reformulate Eq. (3) as a linear least squares problem: $\|\mathbf{A}\boldsymbol{\Delta\theta} - \mathbf{b}\|^2$, which can be solved in closed-form. The parameters are updated as $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \boldsymbol{\Delta\theta}$, and this procedure is repeated until convergence. To improve efficiency, we also adopt the inverse compositional method [10].

NormalFlow offers advantages over ICP by leveraging surface normals for pose estimation. Consider determining the tilt of a flat surface. The rotations of surface normals directly reveal the tilt, whereas estimating tilt using distorted point clouds can be noisy. For a textured ball, pose estimation should rely on the textures. Unfortunately, ICP will focus on registering the global ball shape since the textures minimally affect point locations. In contrast, NormalFlow will estimate pose from textures by matching the diverse normal directions in the textured region.

III. Experiments and Results

In this section, we perform experiments to evaluate the accuracy and speed of the NormalFlow for tracking the motion of 10 different objects. Experiments are conducted on the GelSight Mini tactile sensor [11] with a resized resolution of 320 × 240. This sensor has a 20mm × 15mm sensing area and operates at 25 Hz. We collect seven tracking trials for each object, with contact initiated at different poses shown in Fig. 3. Trials average 10.2 seconds in duration, and ground-truth sensor poses are recorded using a motion capture (MoCap) system.

A. Baseline Methods

We compare NormalFlow with three common point cloud registration methods:

Point-to-Plane ICP [7]: Referred to as ICP in this paper, Point-to-plane ICP is a local registration approach and is commonly used for tactile-based tracking [8] [5] [4]. We utilize the implementation from the Open3D Library [12].

FilterReg [13]: FilterReg is a probabilistic local point cloud registration approach. It is more robust than ICP and has been applied in [14] for tactile registration. We utilize the implementation from the ProbReg Library [15].

FPFH + RANSAC + ICP: Abbreviated as FPFH+RI in this paper, it combines Fast Point Feature Histograms (FPFH) [16] and RANSAC to extract features and register point clouds globally. It then applies ICP to refine the alignment. The approach is applied in [6] for tactile registration. We utilize the implementation from the Open3D Library [12].

B. Tracking Results

For all methods, we estimate and evaluate the 6DoF sensor pose mean absolute error (MAE) for each frame

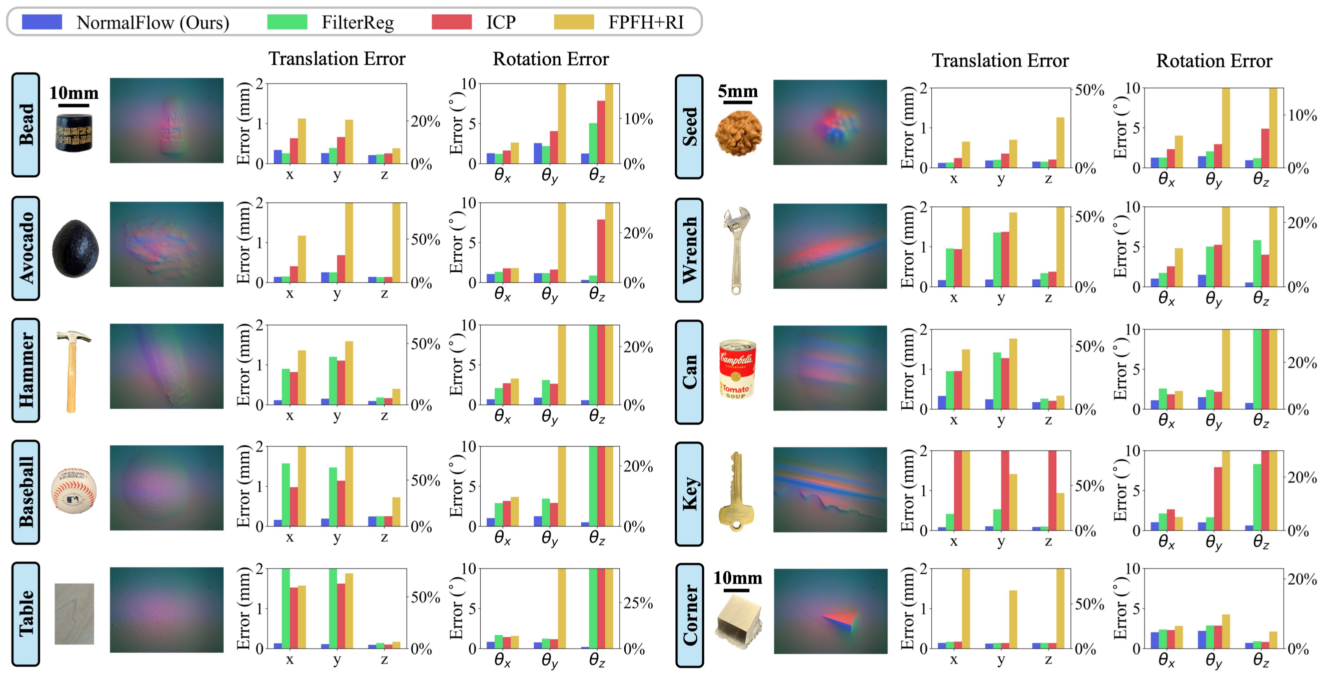

Fig. 4: Tracking results for the 10 objects. For each object: **[left]** the object (scale not shown for common objects) and a sample tactile image (Seed's image slightly blurred to avoid visualization discomfort); **[right]** the 6DoF tracking MAE: left y-axis shows absolute error, right y-axis shows percentage error relative to object movement range.

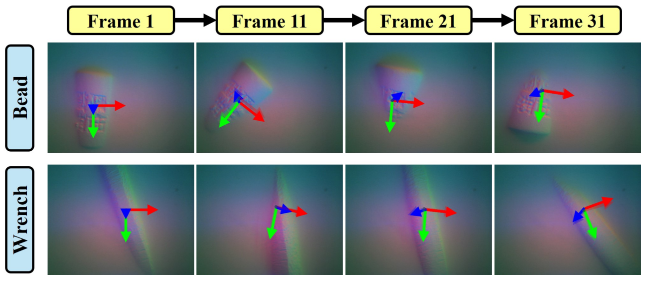

Fig. 5: Example trials on two objects. RGB axes show NormalFlow estimated poses. Transparent RGB axes show true poses, nearly overlapping with NormalFlow poses.

Method	x(mm)	y(mm)	z(mm)	$\theta_x(^\circ)$	$\theta_y(^\circ)$	$\theta_z(^\circ)$
NormalFlow	**0.17**	**0.18**	**0.15**	**1.13**	**1.42**	**0.64**
FilterReg	0.85	1.05	0.20	1.96	2.59	15.4
ICP	1.22	3.44	0.85	2.27	3.30	15.9
FPFH+RI	2.38	1.69	1.26	2.93	36.8	27.8

TABLE I: 6DoF tracking MAE

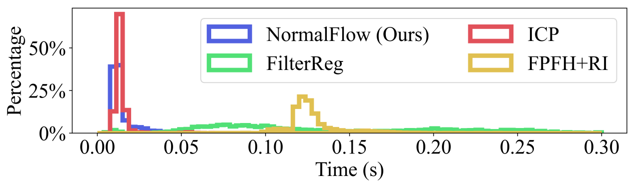

Fig. 6: The runtime histogram. The average runtimes are: NormalFlow (13.9 ms), ICP (13.6 ms), FilterReg (145 ms), and FPFH+RI (127 ms).

relative to the first frame. The tracking result is shown in Fig. 4 and Table I. Two example tracking trials using NormalFlow are shown in Fig. 5. NormalFlow outperforms baselines on all objects, particularly objects with less textures. We find that FPFH+RI often falls into local minima, explaining the challenges of extracting features from tactile point clouds. Meanwhile, ICP consistently performs worse than NormalFlow and FilterReg. While NormalFlow only slightly exceeds FilterReg's performance on highly textured objects (like Avocado and Seed), it significantly outperforms FilterReg on less textured objects (like Wrench and Hammer) by maintaining robust tracking where FilterReg often fails. To the extreme, NormalFlow can robustly track objects like Table, which is considered textureless by human standards.

C. Runtime Analysis

We measure the runtime of all algorithms on a laptop equipped with an AMD Ryzen 7 PRO 7840U CPU without GPU acceleration. The runtime histograms are shown in Fig. 6. The average runtime of NormalFlow is 13.9 ms, closely comparable to ICP at 13.6 ms, and significantly faster than FilterReg at 145 ms and FPFH+RI at 127 ms.

D. Long-horizon Tracking Results

We demonstrate the long-horizon tracking performance of our approach using three objects with varying tracking difficulty: Bead (easy), Wrench (medium), and Table (hard).

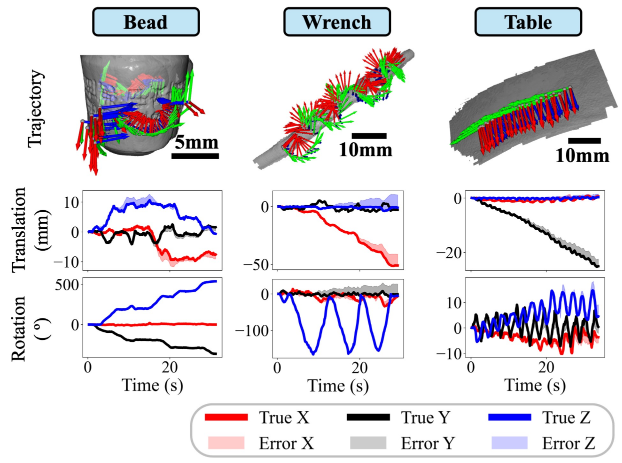

Fig. 7: Long-horizon tracking results. For each column: **[top]** sensor trajectories with poses as RGB coordinate axes; **[bottom]** true 6DoF sensor movements tracked by MoCap (solid lines) and NormalFlow estimation error (shaded area), which is often too small to be seen.

For each object, we conduct a trial where the sensor travels a significant distance from its initial pose. The NormalFlow tracking results (Fig. 7) indicate minimal tracking error even after extensive movement. After rolling the Bead 360 degrees along the y-axis and twisting it 540 degrees along the z-axis, the tracking errors in the y-axis and z-axis remain 2.5 and 1.8 degrees, respectively. Note that determining rolling angles along the y-axis has been previously considered challenging [3].

IV. Application: Tactile-based 3D Reconstruction

We demonstrate the power of NormalFlow by applying it to the task of tactile-based 3D reconstruction. In our experiment, the target object is manually rolled across the GelSight Mini, revealing small surface patches in each tactile frame. NormalFlow tracks the 6DoF pose over time, and when a loop closure is detected, it estimates the relative pose between the two endpoints. These poses are optimized in real time using pose graph optimization. As shown in Fig. 8, our approach produces highly detailed reconstructions of the object surface—details that are often difficult to capture with visual methods. Compared to prior approaches such as [3], our approach delivers significantly improved 3D reconstruction quality. Its success highlights the precision of NormalFlow, where even small errors can cause severe artifacts in the final mesh.

V. Conclusion

In this work, we present NormalFlow, a fast, robust, and accurate 6DoF pose tracking algorithm for vision-based tactile sensors. Because our method tracks objects directly through contact, it also reveals fine-grained contact modes such as sliding and rolling. By directly minimizing discrepancies between surface normal maps instead of point clouds, our approach outperforms baseline methods and

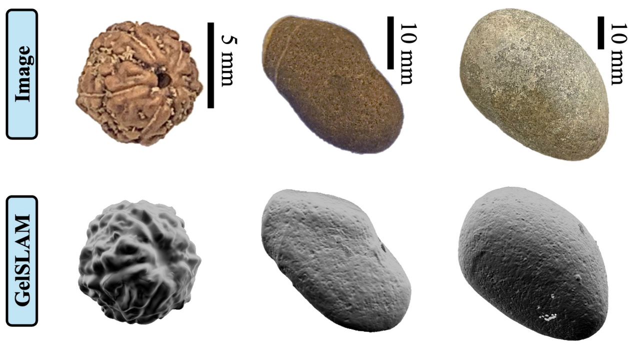

Fig. 8: Tactile-based 3D reconstruction results enabled by NormalFlow tracking.

tracks well even with low-texture objects. We demonstrate the effectiveness of NormalFlow in tactile-based 3D reconstruction. We believe NormalFlow can be broadly applied, enabling new advancements in high-precision perception and control, particularly in dexterous manipulation tasks such as handwriting, soldering, and tool use.

References

[1] O. M. Andrychowicz, B. Baker, M. Chociej, R. Józefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. Weng, and W. Zaremba, "Learning dexterous in-hand manipulation," *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 3–20, 2020.
[2] W. Yuan, S. Dong, and E. H. Adelson, "Gelsight: High-resolution robot tactile sensors for estimating geometry and force," *Sensors*, vol. 17, no. 12, 2017.
[3] J. Zhao, M. Bauza, and E. H. Adelson, "Fingerslam: Closed-loop unknown object localization and reconstruction from visuo-tactile feedback," 2023.
[4] S. Suresh, H. Qi, T. Wu, T. Fan, L. Pineda, M. Lambeta, J. Malik, M. Kalakrishnan, R. Calandra, M. Kaess, J. Ortiz, and M. Mukadam, "Neural feels with neural fields: Visuo-tactile perception for in-hand manipulation," 2023.
[5] P. Sodhi, M. Kaess, M. Mukadanr, and S. Anderson, "Patchgraph: In-hand tactile tracking with learned surface normals," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE Press, 2022, p. 2164–2170.
[6] J. Lu, Z. Wan, and Y. Zhang, "Tac2structure: Object surface reconstruction only through multi times touch," *IEEE Robotics and Automation Letters*, vol. 8, no. 3, pp. 1391–1398, 2023.
[7] Y. Chen and G. G. Medioni, "Object modeling by registration of multiple range images," *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, pp. 2724–2729 vol.3, 1991.
[8] S. Wang, Y. She, B. Romero, and E. H. 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.
[9] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, ser. IJCAI'81. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1981, p. 674–679.
[10] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *International Journal of Computer Vision*, vol. 56, no. 3, pp. 221 – 255, March 2004.
[11] "GelSight Mini," https://www.gelsight.com/gelsightmini/.
[12] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," *arXiv:1801.09847*, 2018.
[13] W. Gao and R. Tedrake, "Filterreg: Robust and efficient probabilistic point-set registration using gaussian filter and twist parameterization," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 11 087–11 096.

[14] M. Bauzá, E. Valls, B. Lim, T. Sechopoulos, and A. Rodriguez, "Tactile object pose estimation from the first touch with geometric contact rendering," in *Conference on Robot Learning*, 2020.
[15] Kenta-Tanaka et al., "probreg," 2019. [Online]. Available: https://probreg.readthedocs.io/en/latest/
[16] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 3212–3217.