Learning Bayes Filter Models for Tactile Localization

Tarık Keleştemur
College of Engineering
Northeastern University

Taşkın Padır
College of Engineering
Northeastern University

Robert Platt
Khoury College of Computer Sciences
Northeastern University

***Abstract*—Tactile localization is a useful method to localize end-effectors of manipulators that do not have accurate position encoders or use soft materials which cannot be modeled accurately. However, making sense of noisy tactile feedback for localization is a challenging yet compelling problem. In this work, we present learnable Bayes filter models that can localize robotic grippers using tactile feedback. We propose a novel observation model that conditions the tactile feedback on visual maps of the environment along with a motion model to recursively estimate the gripper's location. Our models are trained in simulation with self-supervision. We evaluate our method on a tabletop localization task in which the gripper interacts with objects. We report results in simulation generalizing over different sizes, shapes, and positions of the objects.**

I. INTRODUCTION

Tactile perception gives robots the ability to make sense of the physical world by leveraging contact interactions. It is particularly important for manipulation tasks where vision cannot provide useful information about the state of the world. In literature, tactile perception is shown to be crucial for tasks such as grasping, in-hand manipulation, object recognition, and non-prehensile manipulation. In this work, we show that tactile feedback can also be used for the localization of robotic grippers. Although there have been studies that show the use of tactile feedback for object localization, the problem of the gripper localization using tactile feedback is limited.

In the last few years, there has been growing attention on embedding algorithmic priors of planning, control, and state estimation into neural networks. One line of research in this direction is learning sequential Bayes filter models which are shown to outperform arbitrary neural network architectures for state estimation problems in partially observable environments. Jonschkowski and Brock [4] and Karkus et al. [6] proposed differentiable histogram filters and showed that it can outperform LSTM-based networks in environmentss with discretized state spaces. Several improvement have been made improve learnable Bayes filters such as using Particle filter [5] or Kalman filters [3] to handle continuous environments.

In the manipulation literature, tactile localization is often defined as inferring object locations using tactile feedback [1, 8] but we tackle the problem of inferring gripper pose in the world coordinates. The closest work to ours is [7] in which the authors aim to localize a tactile sensor in visual maps using Bayes filtering. A visual-tactile sensor is used to generate SIFT features of the current observation. These local features are then matched with the SIFT features of the visual map to calculate the likelihood probabilities. The major limitation of this work is that for every new map, the pre-processing step of calculating the visual map's features needs to be reperformed.

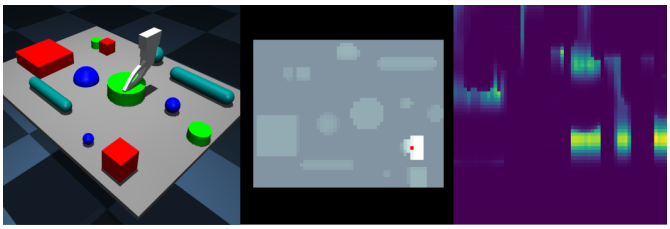

Fig. 1. **Left:** Simulation - **Middle:** Depth Image - **Right:** Belief Map

II. APPROACH

The main objective of this work is to localize a robotic gripper with respect to an image using tactile feedback. We formulate our problem as discrete Bayes filtering as introduced by Thrun [11] which is widely used for state estimation with uncertain models. Bayes filtering maintains a *belief* over state space and recursively update the belief using *observation model* and *motion model*:

$$bel(s_t) = \eta p(o_t|s_t) \sum_{s_{t-1} \in S} p(s_t|s_{t-1}, a_{t-1}) bel(s_{t-1}) \quad (1)$$

where η is the normalization factor, $p(o_t|s_t)$ is the observation model, and $p(s_t|s_{t-1}, a_{t-1})$ is the motion model. These models are parameterized by neural networks and learned from data using gradient descent.

To this end, we have two goals: (1) Learn an observation model that is conditioned on an image, e.g. an image of the tabletop. This observation model would describe the expected tactile feedback as a function of gripper position for a new scene, as conveyed by the image of the tabletop surface, (2) Learn a motion model that transitions belief at each state to the next timestep. We use images of the environment as the map where each pixel is a discrete representation of the gripper's location. The state space is defined as the pixel coordinates of the gripper $s_t = (p_x, p_y) \in \mathbb{Z}^{H \times W}$ where H is the height and W is the width of the image, I. For tactile observations, we use the the last 16 joint positions of gripper fingers received until the current state. The belief space is encoded

as a $H \times W$ matrix where the matrix elements represent the probability values of the belief. Our problem can be formulated as estimating the $bel(s_t)$ of the state given the image of the environment I and the observation o_t. Let $f_M(\cdot)$ be a function that takes the current belief as input and outputs the predicted belief at next timestep, i.e. $f_M(bel(s_{t-1})) = \overline{bel}(s_t)$ and $f_O(\cdot)$ be a function that takes the current observation and image of the environment as input and outputs the likelihood of observing o_t in the image I, i.e. $f_O(o_t, I) = p(o_t|s_t, I)$. We can then write the observation and prediction steps as:

$$bel(s_t) = \eta f_O(o_t, I) \odot f_M(bel(s_{t-1})) \tag{2}$$

A. Bayes Filter Networks

For observation model $f_O(o_t, I)$, we use a modified U-Net architecture developed by Ronneberger et al. [9]. The network consists of 3 modules: (I) *Image Encoder* is a stack of 6 2D convolution layers which takes the depth image and produce latent feature maps, (II) *Observation Encoder* is composed of 3 1D convolution layers and it takes the observation and generate latent features maps, (III) The *Likelihood Decoder* takes both set of feature maps and feed into 6 2D convolutional layers by concatenating the feature maps. The 2D convolutional layers are followed by batch normalization and ReLu activation except for the last layer of the decoder. The 1D convolutions layers are followed by ReLu activation. For the Image Encoder, max pooling layers are used for down sampling and for the Likelihood Decoder, we use transposed convolutional layers for up sampling. The last layer of the decoder has 16 channels representing 16 discrete probability values for each pixel. Each pixel gives the probability value of the gripper being in that pixel given the observation and the depth image. To get pixel-wise probability values, we apply depth-wise softmax and train the network using categorical cross entropy loss. Since the likelihood map refers to a probability mass function, the elements of the map need to be positive and sum to one. To realize this, we normalize the output of the network over pixels.

The motion model takes the previous belief as the input and makes predictions for the next timestep. Since the finger only moves with linear velocity, our motion model is not conditioned on actions. We define our motion model $f_M(\cdot)$ as a 2D convolution operation where the kernel weights $w_M \in R^{3\times3}$ represent the transition probability function, thereby, the elements of the kernel need to be positive and sum to one. This is enforced with softmax normalization over the weights.

B. Data Collection

We generate a dataset of state transitions and collect environment images, observations and ground truth likelihood maps by rolling out simulations. A simulation environment is developed using the MuJoCo physics engine developed by Todorov et al. [12] to collect the dataset. The environment consists of a free-floating gripper [10], a table and objects on the table (see Figure 1). A top-down facing depth camera is positioned over the table. The gripper moves from one edge of the table with a linear motion up to the opposing edge of the table. As explained in Section II, the state space is the coordinates of the gripper in the camera frame. In order to find the pixel coordinates of the gripper, we first transform the pose of the gripper base to the camera frame and then project it into pixel coordinates: $p = M_{int} M_{ext} P_w$ where M_{int} and M_{ext} are intrinsic and extrinsic camera matrices, respectively, and $p = (p_x, p_y)$ is the vector of pixel coordinates of the gripper. The joint angles of the gripper are used as the tactile observations. The gripper used in this work has hydrostatic linear actuators which allows us to set the finger joint stiffness to a low value. This way, the gripper can interact with objects without moving them. To generate the dataset, we first sample an environment configuration by sampling object positions and and sizes. Then, the gripper traverse over the each pixel rover and collect the observation for each state. To train the network, we generate ground truth likelihood maps by calculating the distance of the observations to each other. These distances are then normalized between 0 and 1 to get the probability values. Finally, we discretize probability values into 16 categories to use cross-entropy loss.

III. Experiments

Both of the networks are trained with data collected using the simulation environment by self-supervision. The trained model are combined for recursive state estimation. To evaluate the localization success, we perform the filtering on unseen episodes. We show results on two sets of objects: (I) Primitive objects including squares, spheres, capsules, and cylinders, (II) YCB objects [2]. Note that the networks are only trained on the primitive object set to show that it can generalize over new objects. Our performance metric is the Manhattan distance between the true state and predicted state at the end of a trajectory. For the experiments, we generate a randomized scene and run 64 episodes and take the mean error. For primitive objects, we achieve a mean error of 1.15 and for YCB objects we achieve a mean error of 5.07.

IV. Conclusions

In this work, we have addressed the problem of tactile localization where the goal is to localize a robotic gripper using tactile feedback. We formulate the localization problem as a recursive Bayes filtering problem and learn the filter models from data. A self-supervised and simulation-based data collection procedure is introduced to collect contact interactions between the environment and the gripper. We showed that in addition to successful localization with unseen object configurations and sizes, our approach can also localize with novel objects. The main drawback of our method is the assumption that the gripper does not move objects. To mitigate this problem, we would like to extend the localization formulation to track the objects as well. Our future work also includes combining the proposed localization method with policy learning to perform manipulation tasks by jointly learning the localization and planning.

REFERENCES

[1] Maria Bauza, Oleguer Canal, and Alberto Rodriguez. Tactile mapping and localization from high-resolution tactile imprints. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3811–3817. IEEE, 2019.

[2] Berk Calli, Aaron Walsman, Arjun Singh, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. Benchmarking in manipulation research: The ycb object and model set and benchmarking protocols. *arXiv preprint arXiv:1502.03143*, 2015.

[3] Tuomas Haarnoja, Anurag Ajay, Sergey Levine, and Pieter Abbeel. Backprop kf: Learning discriminative deterministic state estimators. In *Advances in Neural Information Processing Systems*, pages 4376–4384, 2016.

[4] Rico Jonschkowski and Oliver Brock. End-to-end learnable histogram filters. In *Workshop on Deep Learning for Action and Interaction at NIPS*, December 2016.

[5] Rico Jonschkowski, Divyam Rastogi, and Oliver Brock. Differentiable particle filters: End-to-end learning with algorithmic priors. In *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, June 2018. doi: 10.15607/RSS.2018.XIV.001.

[6] Peter Karkus, David Hsu, and Wee Sun Lee. Qmdp-net: Deep learning for planning under partial observability. In *Advances in Neural Information Processing Systems*, pages 4694–4704, 2017.

[7] Shan Luo, Wenxuan Mou, Kaspar Althoefer, and Hongbin Liu. Localizing the object contact through matching tactile features with visual map. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3903–3908. IEEE, 2015.

[8] Robert Platt, Frank Permenter, and Joseph Pfeiffer. Using bayesian filtering to localize flexible materials during manipulation. *IEEE Transactions on Robotics*, 27(3): 586–598, 2011.

[9] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[10] Eric Schwarm, Kevin M Gravesmill, and John P Whitney. A floating-piston hydrostatic linear actuator and remote-direct-drive 2-dof gripper. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7562–7568. IEEE, 2019.

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

[12] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012.