Project Report: Tactile Exploration of Objects

Jan Rüggemeier
Chair of Robotics Artificial Intelligence and
Real-time Systems
Technical University of Munich
jan.rueggemeier@tum.de

Daniel Strauß
Chair of Robotics, Artificial Intelligence and
Real-time Systems
Technical University of Munich
daniel.strauss@tum.de

***Abstract*—Robot grasping relies on accurate 3D models from sensory data of depth cameras, cameras or tactile exploration. Related work relies often on point cloud data from cameras in combination with sparse tactile data. We explore a novel approach to tactile exploration with only sparse tactile data availabe. For this task we create our own dataset, train a reconstruction network for shape prediction and enhance the tactile exploration with reinforcement learning. The results show an increase in performance in comparison to a random policy.**

***Index Terms*—tactile exploration, robot grasping, shape prediction**

I. Objective

In robotics, a challenge lies in effectively controlling robotic hands for object grasping. An important aspect of this challenge involves generating accurate 3D models of the objects. This task typically relies on sensory data from depth cameras, cameras, and tactile exploration. The primary aim of this project is to investigate a new approach to enhance tactile exploration. Specifically, this approach involves implementing a reinforcement learning (RL) algorithm designed to optimize the selection of exploration points based on previous tactile exploration outcomes. Initially, the focus will be on achieving this optimization for 2D objects. Potential further improvements may involve extending the methodology to 3D objects and integrating visual data to improve the reconstruction process of both 2D and 3D models.

II. Related Work

There have been several previous studies about shape prediction for robotic grasping tasks. Humt et al. [5] perform the shape prediction based on data from a depth camera and grasp prediction for a good grip on the object without further tactile exploration. Watkins et al. 2019 [1] combine data from a depth camera and sparse tactile points for this shape prediction task. Even though they include tactile exploration data for shape prediction, it does not include a tactic for choosing suitable new grasping points to improve its prediction. In [6] and [4], tactile exploration is improved by calculating the locations of highest uncertainty and choosing that location for the next tactile exploration. In [6], uncertainty is quantified using Eikonal loss, and in [4], using a reconstruction network. Another approach aiming to optimize the tactile exploration points was performed in [3] using RL. In [3] - unlike at this work - before tactile exploration, a point cloud of depth data is available. The RL algorithm, optimizing the locations of tactile exploration, would at each timestep reward a higher information gain. In that study, the information gain was inferred from a probabilistic model, which inferred uncertainty from previous tactile exploration of that object but was independent of the knowledge of the shape prediction network. As our work does not include an initial point cloud, another shape prediction network may be used. Additionally, the reward function might be improved if information about information gain can be inferred from the shape prediction network similar to [4].

III. Technical Outline

In this project we limited ourselves to 2D objects for an easier implementation. The project consisted of these three main parts.

- **Creation of the Dataset**(Sec. IV: In this part, we created a dataset by converting 3D models to 2D images and prepared a representation of tactile points for the reconstruction network and a representation of the outline for the reinforcement learning agent.
- **Reconstruction from Tactile Points**(Sec. V: Then we trained a neural network to reconstruct a shape from an arbitrary set of tactile points.
- **Optimized tactile exploration** with reinforcement learning (Sec. VI: In the last part, we train with RL agents to find optimal tactile points to feed into the reconstruction network, such that the reconstruction is improved.

IV. Dataset

We used 3D models from the Mug and the Bottle datasets of ShapeNet to train the model. To convert the 3D models into 2D models, we used PyRender to project the models from the same perspective to a 2D map.

Thereby, we obtained 712 2D images of resolution $1 \times 256 \times 256$, which we split into training- (80%), validation- (10%) and test-set (10%). The image would then contain binary values, where pixel i, j would be one if, in the projection, the object would be at the location at this pixel. After splitting the images into different sets, we applied 5 random rotations to each image to avoid all mugs and bottles being oriented identically, ensuring a more realistic data distribution. It is important to remark that these rotations had to be applied after the split into train-, validation-, and test sets to ensure the meaningfulness

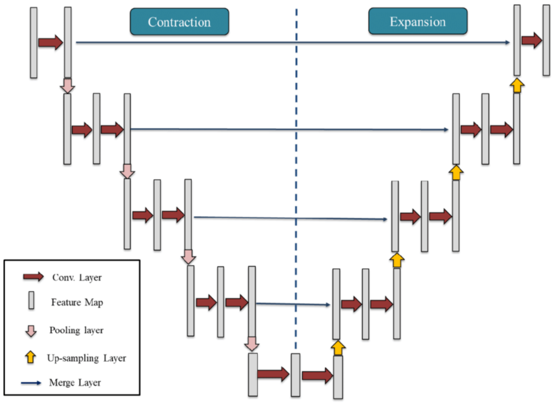


Fig. 1: The U-Net Architecture (taken from [2])

of validations using the validation and test set. Then, for each image, its outline was calculated, and from the outline, 1 to 10 random tactile points were randomly sampled to train the reconstruction net with them later. Outline and random tactile points were stored in 2D binary images of the same resolution.

The code was designed in such a way, that the datatset size and distribution can be modified. Additional model classes from ShapeNet can be easily added and converted by providing the respective ShapeNet url.

V. Reconstruction Net

The reconstruction network should reconstruct a 2D model from an arbitrary set of 1 to 10 tactile points. We chose the U-Net architecture as we decided to reduce the problem to do a segmentation task, and the U-Net architecture is a simple and widely used architecture for this task.

The U-Net receives as input a $1 \times 256 \times 256$ image, which encodes the positions of the tactile points. The training label is the corresponding $1 \times 256 \times 256$-image, which represents the 2D projection of the model. The generation of these images is described in Section IV. The idea of the $1 \times 256 \times 256$ output is that the U-Net shall produce a 2-category segmentation map, in which each pixel describes the probability of the object being there. To obtain such a probabilistic description, the σ-activation is applied to the last layer, and the entropy loss is used for training.

Before fine-tuning the model, we first tested the potential of the U-Net by seeing whether it was capable of overfitting the test set models on a fixed number of 60 tactile points. Afterwards, we changed the code to allow for adjustable U-net depth and channel number (The amount n of channels in the first layer was variable, and then it would be scaled by two for each contracting block). Then, we used the *Ray* library to fine-tune our model for the parameters learning rate, batch size, depth, and number of channels with the Bayesian Optimization with Hyperband (BOHB) algorithm. We used the BOHB algorithm because of its efficiency and robustness. We ran the parameter search for 100 samples and 10 epochs each to increase our chances in finding optimal parameters. The search resulted in the following choice of hyperparameters (see Tab. I:

TABLE I: BOHB hyperparameter results

Hyperparameter	Value
Batch size	4
Learning rate	2.074e-5
Depth	6
Channels	128

We chose the *Jaccard index* as an additional metric, as the accuracy over the whole image would be high by default as the shapes are concentrated in a small image area. The *Jaccard index* better evaluates the performance by quantifying the overlap of the actual shape (set of pixels L) with the predicted shape (set of pixels O):

$$J(O, L) = \frac{O \cap L}{O \cup L}$$

Finally, the performance of the reconstruction network is outlined in table II. The performance on the test set is nearly the same as for the evaluation set, which indicates a good data distribution in our dataset splits.

TABLE II: Performance of reconstruction network

Dataset	Loss	Jaccard-Index
Train	0.0276	82.45
Eval	0.0350	77.37
Test	0.0350	77.48

VI. Reinforcement Learning

Finally, we used our dataset and our reconstruction net to train a reinforcement algorithm to select optimal grasping points for tactile exploration of unknown object shapes. Rays act as an approximation of grasping approaches, and the intersection with the object outline is the new grasping point. We selected the *Stable Baselines3* library to set up an environment and perform training.

A. Environment

To improve compute time and simplify the setup, we confined ourselves to the discrete pixel space of the input image and used *scikit-image* for efficient operations on the pixel level. As every shape is located in the center of the input images and is confined to a small region, we construct a circle with a radius of 127px around the middle pixel closest (0,0). We then defined a two-dimensional discrete action space, which returns two indices *alpha* and *beta* corresponding to two pixels on the circle. The pixel-line between these pixels is the casted ray of our model.

After initialization, our environment performs the following steps until termination after ten steps:

1) The pixel line between *alpha* and *beta* is constructed, and the intersection point with the object shape closest

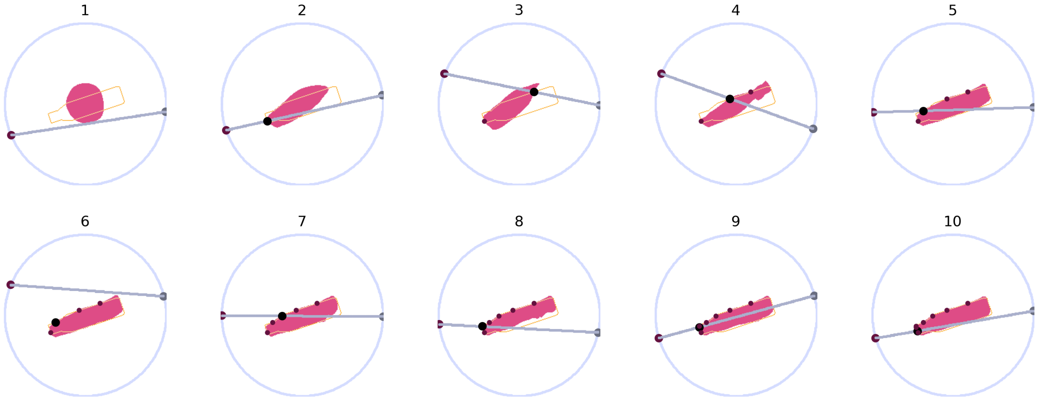


Fig. 2: Individual steps of Total-policy on test set sample.

to *alpha* is added to the list of grasping points (if it exists).

2) The list of grasping points is converted to a grasping point image, where every grasping point pixel has the value 1. This image is inferred by our reconstruction net with the current label image. It returns an image of the predicted shape, the loss, and the *Jaccard index*.
3) The prediction image and the grasping point image are combined into a two-channel observation, and the reward function is calculated.

After termination, the environment is reset, and a random label image is chosen from the current dataset.

B. Training

We chose *PPO* with the *CNNpolicy* as the algorithm for the training. Because of its simplicity, stability, and sample efficiency, it is a good choice for our first training approaches in the field of reinforcement learning. When applying the algorithm in its base configuration of *Stable Baselines3* with the loss as a basic reward, it converged to a single action without a good score. This was fixed by setting the entropy coefficient from 0 to 0.01.

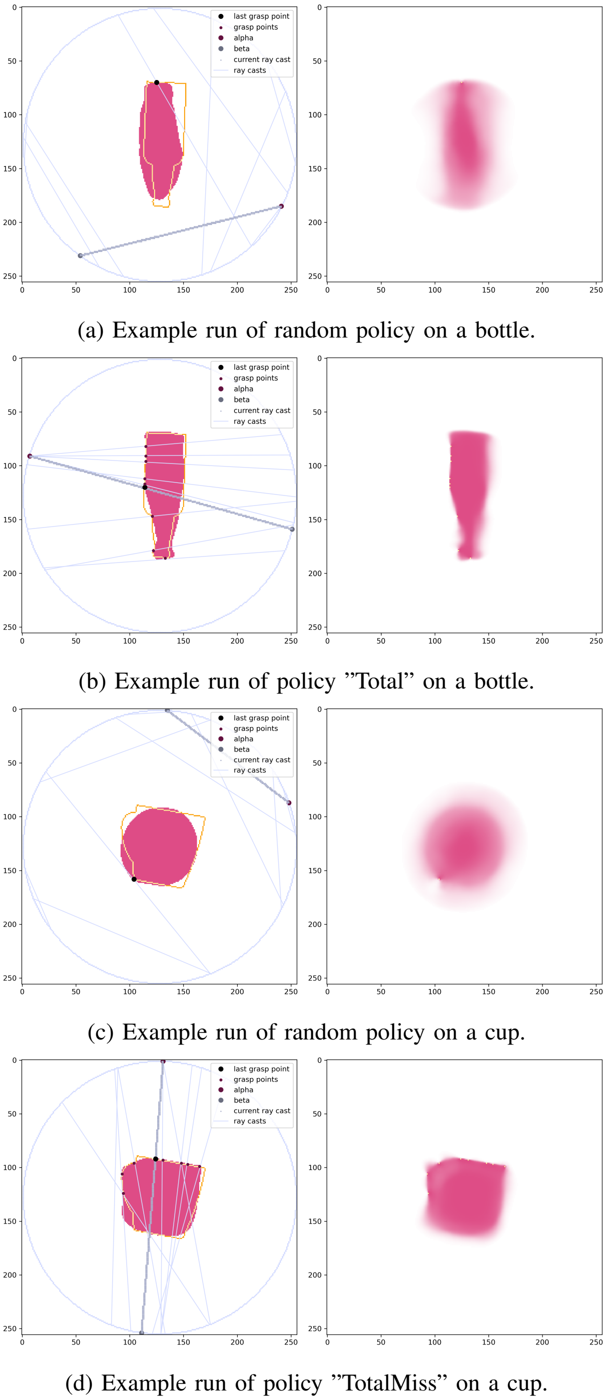
We trained PPO on a total of four different training approaches with differing reward functions. Every reward function uses the *Jaccard index* as the metric:

- **Diff**: The reward function returns the difference of the metric with the previous steps as a reward. Additionally, misses of the object shape and hitting a previous grasping point again are penalized with a -10 reward. We trained this reward function for 500k steps, after which it converged to a relatively constant average reward. Each iteration on a training sample is terminated after ten steps to simulate the ten grasping approaches the reconstruction net was trained on.
- **Total**: This reward function and training approach is the same as *Diff* but returns the current metric value with the same penalties.
- **TotalMiss**: It also uses the current metric value as a reward, but the environment is set to terminate the current iteration when ten grasping points are calculated and not when ten steps have been performed. Additionally, no penalty is enforced. After 500k training steps, the algorithm is trained for an additional 500k training steps with the *Total* training approach.
- **preDiffMiss**: This approach is the similar to *TotalMiss*, but with the difference of the metric with the previous step as reward. It was only trained for the first 500k steps without penalties. The second 500k training with the *Diff* approach was omitted due to hardware limitations and time constraints. Even though this approach is unfinished, we still evaluated it to compare it to the other three approaches and the random policy.

VII. Results

In Fig. 3 we can see two runs (bottle and mug) by a random policy and our trained *Total* and *TotalMiss* policy on the same sample from the test set. We defined the random policy to sample random values from the action space without any further constraint. The random policy does not hit the bottle or mug most of the time and performs worse on this sample than the trained policy. Still, the reconstruction net already predicts a similar area close to the actual shape based on only one tactile point, resulting in a high metric. Additionally, the shape prediction on the right side is contained in circle around the middle point. It seems, that the network learned a circular boundary, where every training sample was contained in. It also seems to utilize the symmetry of the bottle and mug objects in our dataset as can be seen in the lower right plot, where all tactile points lie on the left side of the bottle.

Both trained policies are hitting the outline with nearly every ray, resulting in an overall better prediction of the actual shape. The policies seems to limit the actions parralel rays from one side to the other. Also, the position of the *alpha* and *beta* seem to coincide for many rays and do not show a lot of variation. The policy seems to ignore the uncertain regions of the prediction image and is converged on casting the rays in one direction. An extension of the reward function to include rewards based on the uncertainty of the region where the



(a) Example run of random policy on a bottle.

(b) Example run of policy ”Total” on a bottle.

(c) Example run of random policy on a cup.

(d) Example run of policy ”TotalMiss” on a cup.

Fig. 3: Example runs for different policies for 10 steps on two distinct test set samples. The *left* side shows the actions (rays and grasping points) and the threshold of the prediciton for reward calculation in *pink*. The *right* side displays the observation space for the policies. Pink shows the probabilistic output of the reconstruction net and yellow shows the grasping points.

new grasping point lies in, might improve the policy. The reconstrction net shows a noticable, distinct border line in regions with many grasping points (see Subfig. 3b).

In Fig. 2 we can see the individual rays and predictions for each step of the trained Total-policy on a test set sample. It still misses in some instances, for example at step 1 and 6. This is most likely due to the fact, that this sample shape is oriented in the same direction the policy casts the ray. Still, sfter the first miss it adapts the ray to hit the object, resulting in a better predicitoon than the circular standard prediction. In the beginning, the reconstruction net predicts the bottle to be oriented differently, but with more grasping points achieved by the policy, the prediction converges close to the label. This shows the potential of improvement in the shape prediciton with reinforcement learning.

We evaluated all of our four training approaches and a random policy on the test set. For the first evaluation we chose the inital setting of this project and applied each policy on every of the 356 test set samples. For every individual step we accumulated the *Jaccard index* value and calculated the mean over every sample (see Fig. 4). All of our four policies outperform the random policy and result in an average metric value of ca. 80 at step ten, while the random policy only reaches an average value of about 60. Notably, enforcing a penalty on misses and doubles increases the performance gain in the first steps, as *preDiffMiss* without any penalty has a lower slope than the other three trained policies.

In order to explore, whether the performance gain through the trained policies stems from an overall better selection of grasping points or just hitting the object shape more regularily, we also adapt our environment to only terminate, when ten grasping points are reached. Therefore, we can compare the policies, with the random policy having missed the object. We also accumulate the mean metric values, but for each attained grasping point instead of step (see Fig. 5). While the three policies *Total*, *Diff*, and *TotalMiss* show nearly the same performance to the previous setting, the *preDiffMiss* policy increases slightly in performance. The random policy outperforms our policies by reaching an average metric value of about 90 after ten grasping points. This shows, that the policies outperform the random policy in the original setting mainly by hitting the object outline more regularily. With this setting the random policy seems to gather better grasping points by casting rays from different directions on the object instead of the limited, one-directional rays of the other policies. Nonetheless, the evaluation of the test set on ten grasping points took more than three times as long for the random policy than the other policies.

VIII. Conclusion & Outlook

In this project, we evaluated the option of using reinforcement learning to enhance tactile exploration for object shape reconstruction. Our trained policies outperformed the random policy, mainly due to the reason that our policy reduced the the chance of missing the object. Even though we obtained a higher metric than the random policy for termination after ten steps, we did not obtain a higher metric per individual grasping point. This reveals the potential for further improvements

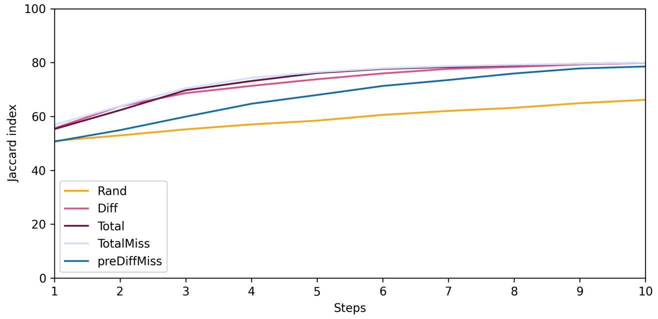


Fig. 4: Policies mean metric per step evaluated over complete test set

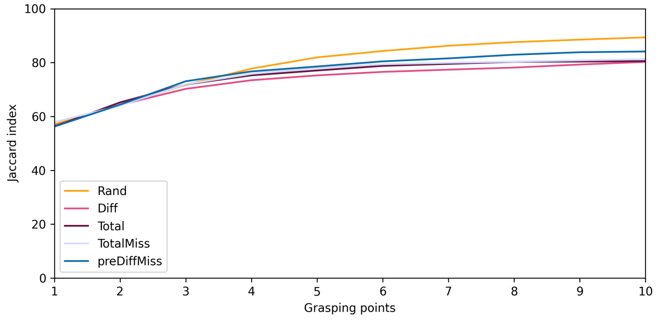


Fig. 5: Policies mean metric per grasp point evaluated over complete test set

in future work. One improvement may yield a more direct tackling of this weakness in the RL policy, for example, by adapting the reward function to positivley reward grasping points in areas of the predicition image without sharp edges ('fuzzy' areas) and exploring other RL algorithms. Further improvements may arise from training on larger datasets, which can simply be generated by sampling more random tactile point sets from the 2D images we have. A further future improvement may arise from extending the RL observation space, for example, by adding missed rays explicitely.

References

[1] D. Watkins-Valls, J. Varley, and P. Allen, "Multi-modal geometric learning for grasping and manipulation," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 7339–7345. DOI: 10.1109/ICRA.2019.8794233.

[2] F. Zabihollahy, "Deep learning methods for abnormality detection and segmentation in computed tomography and magnetic resonance images," Ph.D. dissertation, Aug. 2020. DOI: 10.13140/RG.2.2.26378.70085.

[3] S. Jiang and L. L. Wong, "Active tactile exploration using shape-dependent reinforcement learning," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 8995–9002. DOI: 10.1109/IROS47612.2022.9982266.

[4] L. Rustler, J. Lundell, J. K. Behrens, V. Kyrki, and M. Hoffmann, "Active visuo-haptic object shape completion," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5254–5261, Apr. 2022, ISSN: 2377-3774. DOI: 10.1109/lra.2022.3152975. [Online]. Available: http://dx.doi.org/10.1109/LRA.2022.3152975.

[5] M. Humt, D. Winkelbauer, U. Hillenbrand, and B. Bäuml, "Combining shape completion and grasp prediction for fast and versatile grasping with a multi-fingered hand," in *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*, IEEE, 2023, pp. 1–8.

[6] L. Rustler, J. Matas, and M. Hoffmann, "Efficient visuo-haptic object shape completion for robot manipulation," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 3121–3128. DOI: 10.1109/IROS55552.2023.10342200.