Regrasping using Tactile Perception and Supervised Policy Learning

Yevgen Chebotar* **Karol Hausman*** **Oliver Kroemer** **Gaurav S. Sukhatme** **Stefan Schaal**
University of Southern California
Los Angeles, CA 90089, USA

Introduction

Robust and stable grasping is one of the key requirements for successful robotic manipulation. Although, there has been a lot of progress in the area of grasping [1], the state-of-the-art approaches may still result in failures. Ideally, the robot would detect failures quickly enough to be able to correct them. In addition, the robot should be able to learn from its mistakes to avoid similar failures in the future. To address these challenges, we propose using early grasp stability prediction during the initial phases of the grasp. We also present a machine learning method that is able to learn a regrasping behavior that corrects failed grasps based on tactile perception and improves over time.

In our previous work [2], we presented a first step towards an autonomous regrasping behavior using spatio-temporal tactile features and reinforcement learning. We were able to show that simple regrasping strategies can be learned using linear policies if enough data is provided. However, these strategies do not generalize well to other classes of objects than those they were trained on. The main reason for this shortcoming is that the policies are not descriptive enough to capture the richness of different shapes and physical properties of the objects. A potential solution to learn a more complex and generalizable regrasping strategy is to employ a more complex policy class and gather a lot of real-robot data with a variety of objects to learn the policy parameters. The main weakness of such a solution is that, in addition to requiring large amounts of data, these complex policies often result in the learner becoming stuck in poor local optima [3]. In this paper, we propose learning a complex high-dimensional regrasping policy in a supervised fashion. Our method uses simple linear policies to guide the general policy to avoid poor local minima and to learn the general policy from smaller amounts of data.

The idea of using supervised learning in policy search has been used in [4], where the authors use trajectory optimization to direct the policy learning process and apply the learned policies to various manipulation tasks. A similar approach was proposed in [5], where the authors use deep spatial autoencoders to learn the state representation and unify a set of linear Gaussian controllers to generalize for the unseen situations. In our work, we use the idea of unifying simple strategies to generate a complex generic policy. Here, however, we use simple linear policies learned through reinforcement learning rather than optimized trajectories as the examples that the general policy can learn from.

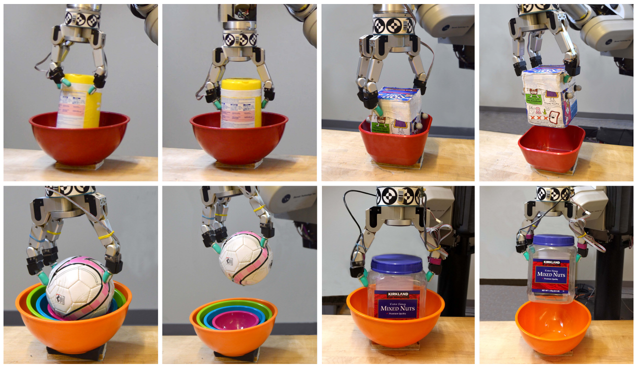

Figure 1: Objects and experimental setup used for learning the grasp stability predictor and the regrasping behavior. *Top-left:* the cylinder. *Top-right*: the box. *Bottom-left:* the ball. *Bottom-right:* the novel object.

Technical Approach

Grasp Stability Prediction with Spatio-Temporal Tactile Features To describe a time series of tactile data, we employ spatio-temporal feature descriptors extracted using Spatio-Temporal Hierarchical Matching Pursuit (ST-HMP) that have been shown to have high performance in temporal tactile data classification tasks [6]. ST-HMP is based on Hierarchical Matching Pursuit (HMP), which is an unsupervised feature-extraction method used for images [7]. In ST-HMP, the tactile information is aggregated both in the spatial and the temporal domains. This is achieved by constructing a pyramid of spatio-temporal features at different coarseness levels, which provides invariance to spatial and temporal deviations of the tactile signal.

In the spatial domain, the dictionary is learned and the sparse codes are extracted from small tactile image patches. To encode data as sparse codes, HMP learns a dictionary of codewords using the common codebook-learning method K-SVD [8]. Given a set of N H-dimensional observations (e.g.

[*] Both authors contributed equally to this work

image patches) $Y = [y_1, \dots, y_N] \in R^{H \times N}$, HMP learns a M-word dictionary $D = [d_1, \dots, d_M] \in R^{H \times M}$ and the corresponding sparse codes $X = [x_1, \dots, x_N] \in R^{M \times N}$ that minimize the reconstruction error between the original and the encoded data:

$$\min_{D,X} \|Y - DX\|_F^2 \text{ s. t. } \forall m \, \|d_m\|_2 = 1 \text{ and } \forall i \, \|x_i\|_0 \leq K,$$

where $\|\cdot\|_F$ is a Frobenius norm, x_i are the sparse vectors, $\|\cdot\|_0$ is a zero-norm that counts number of non-zero elements in a vector, and K is the sparsity level that limits the number of non-zero elements in the sparse codes. The resulting sparse codes are aggregated using spatial max-pooling.

After computing the HMP features for all tactile images in the time series, pooling is performed on the temporal level by constructing a temporal pyramid. The tactile sequence is divided into sub-sequences of different lengths. For all subsequences, the algorithm performs max-pooling of the HMP features resulting in a single feature descriptor for each subsequence. Combined with spatial pooling, this results in a spatio-temporal pooling of the sparse codes.

Finally, the features of all the spatio-temporal cells are concatenated to create a single feature vector F_P for the complete tactile sequence: $F_P = [C_{11}, \dots, C_{ST}]$, where S is the number of spatial cells and T is the number of temporal cells. After extracting the ST-HMP feature descriptor from the tactile sequence, we use a linear Support Vector Machine (SVM) to learn a classifier for the grasp stability prediction [6].

Using multiple levels in the spatial and temporal pyramids of ST-HMP increases the dimensionality of tactile features substantially. When combined with learning regrasping behaviors for multiple objects, this approach leads to a large number of parameters to learn for the regrasping mapping function, which is usually a hard task for policy search algorithms [3]. Thus, in this work, we add several modifications to make this process feasible. In particular, we divide the learning process into two stages: i) learning linear policies for individual objects and ii) learning a high-dimensional policy to generalize between objects.

Learning Linear Regrasping Policies for Individual Objects Once a grasp is predicted to fail by the grasp stability predictor, the robot has to place the object down and regrasp it using the information acquired during the initial grasp. In order to achieve this goal, we learn a mapping from the tactile features of the initial grasp to the grasp adjustment, i.e. the change in position and orientation between the initial grasp and the regrasp. The parameters of this mapping function for individual objects are learned using reinforcement learning. We define the policy $\pi(\boldsymbol{\theta})$ as a Gaussian distribution over mapping parameters $\boldsymbol{\theta}$ with a mean $\boldsymbol{\mu}$ and a covariance matrix $\boldsymbol{\Sigma}$. To reduce the dimensionality of the input features, we perform a principal component analysis (PCA) [9] on the ST-HMP descriptors and use only the largest principal components. The mapping function is a linear combination of these PCA features: $(x, y, z, \alpha, \beta, \gamma) = \mathbf{W}\boldsymbol{\phi}$ with $\mathbf{W} \in \mathbb{R}^{6 \times n}$ and $\boldsymbol{\phi} \in \mathbb{R}^n$, where $\mathbf{W}$ contains the learned weights $\boldsymbol{\theta} = (w_{x,1}, \dots, w_{x,n}, \dots, w_{\gamma,n})$ of the features $\boldsymbol{\phi}$, and n is the number of principal components. The reward $R(\boldsymbol{\theta})$ is computed by estimating the success of the adjusted grasp using the grasp stability predictor. For optimizing the linear policy for individual objects we use the relative entropy policy search (REPS) algorithm [10]. The main advantage of this method is that, in the process of reward maximization, the loss of information during a policy update is bounded, which leads to a better convergence behavior.

The goal of REPS is to maximize the expected reward $J(\pi)$ of the policy π subject to bounded information loss between the previous and updated policy. Information loss is defined as the Kullback-Leibler (KL) divergence between the two policies. Bounding the information loss limits the change of the policy and hence, avoids sampling too far from unexplored policy regions. Let $q(\boldsymbol{\theta})$ be the old policy and $\pi(\boldsymbol{\theta})$ be the new policy after the policy update. We formulate a constrained optimization problem:

$$\max_{\pi} J(\pi) = \int \pi(\boldsymbol{\theta}) R(\boldsymbol{\theta}) \, d\boldsymbol{\theta}$$

$$\text{s. t.} \int \pi(\boldsymbol{\theta}) \log \frac{\pi(\boldsymbol{\theta})}{q(\boldsymbol{\theta})} \, d\boldsymbol{\theta} \leq \epsilon,$$

where, as mentioned before, $J(\pi)$ is the total expected reward of using the policy $\pi(\boldsymbol{\theta})$. The additional constraint bounds the KL-divergence between the policies with the maximum information lost set to ϵ. The updated policy is proportional to the old policy:

$$\pi(\boldsymbol{\theta}) \propto q(\boldsymbol{\theta}) \exp\left(\frac{R(\boldsymbol{\theta})}{\eta}\right).$$

Therefore, we are able to compute the new policy parameters with a weighted maximum-likelihood solution. The weights are equal to $\exp\left(R(\boldsymbol{\theta})/\eta\right)$, where the rewards are scaled by the parameter η. By decreasing η one gives larger weights to the high-reward samples. An increase of η results in more uniform weights. The parameter η is computed according to the optimization constraints by solving the dual problem.

Given a set of policy parameters $\{\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_N\}$ and corresponding episode rewards, the policy update rules for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ can be formulated as follows [3]:

$$\boldsymbol{\mu} = \frac{\sum_{i=1}^N d_i \boldsymbol{\theta}_i}{\sum_{i=1}^N d_i}, \quad \boldsymbol{\Sigma} = \frac{\sum_{i=1}^N d_i \left(\boldsymbol{\theta}_i - \boldsymbol{\mu}\right)\left(\boldsymbol{\theta}_i - \boldsymbol{\mu}\right)^\top}{\sum_{i=1}^N d_i}$$

$$\text{with } d_i = \exp\left(R(\boldsymbol{\theta})/\eta\right).$$

Learning a General Regrasping Policy After the individual linear policies have been learned, we train a larger high-dimensional policy in a supervised manner using the outputs of the individual policies. This is similar to the guided policy search approach proposed in [11]. In our case, the guidance of the general policy comes from the individual policies that can be efficiently learned for separate objects. As the general policy class we choose a neural network with a large number of parameters. Such a policy has enough representational richness to incorporate regrasping behavior for many different objects. However, learning its parameters directly using reinforcement learning requires a large number of examples, whereas supervised learning with already learned individual policies speeds up the process significantly.

To generate training data for learning the general policy, we sample grasp corrections from the already learned individual policies using previously collected data. Input features and resulting grasp corrections are combined in a "transfer" dataset, which is used to transfer the behaviors to the general policy. In order to increase the amount of information provided to the general policy, we increase the number of input features by extracting a larger number of PCA components from the ST-HMP features. Using different features in the general policy than in the original individual policies is one of the advantages of our setting. The individual policies provide outputs of the desired behavior, while the general policy can have a different set of input features.

Experimental Results

In our experiments, we use a Barrett arm and hand that is equipped with three biomimetic tactile sensors (BioTacs) [12]. For extracting ST-HMP features, the BioTac electrode values are laid out in a 2D tactile image according to their spatial arrangement on the sensor.

Evaluation of grasp stability prediction The ST-HTMP features use a temporal window of $650ms$ before and $650ms$ after starting picking up the object. Our goal is to determine early in the lifting phase if the grasp is going to fail. In this manner, the robot can stop the motion early enough to avoid displacing the object, and hence, it can regrasp it later. We evaluate our approach on three objects: a cylindrical object, a box and a ball. We perform a 5-fold cross-validation on 500 grasp samples for each object. The robot achieves a grasp classification accuracy of 90.7% on the cylinder, 82.4% on the box and 86.4% on the ball.

Learning individual linear regrasping policies After learning the grasp stability predictor, we evaluate the regrasping algorithm for individual policies. First, we evaluate individual regrasping policies. The robot performs a randomly generated top grasp using the force grip controller [13], and lifts the object. At the final stage of the experiment, the robot performs extensive shaking motions in all directions to ensure that the grasp is stable. The robot uses the stability prediction to self-supervise the learning process.

To evaluate the results of the policy search, we perform 100 random grasps using the final policies on each of the objects that they were learned on. The robot has three attempts to regrasp each object using the learned policy. Table 1 shows the percentage of successful grasps on each object after each regrasp. Already after one regrasp, the robot is able to correct the majority of the failed grasps, increasing the success rate of the grasps from 41.8% to 83.5% on the cylinder, from 40.7% to 85.4% on the box and from 52.9% to 84.8% on the ball. Moreover, allowing additional regrasps increases this value to 90.3% for two and 97.1% for three regrasps on the cylinder, 93.7% and 96.8% on the box, and to 91.2% and 95.1% on the ball. These results indicate that the robot is able to learn a tactile-based regrasping strategy for individual objects.

Object	Individual policies (# regrasps)				Combined policy
	0	1	2	3	
Cylinder	41.8	83.5	90.3	97.1	92.3
Box	40.7	85.4	93.7	96.8	87.6
Ball	52.9	84.8	91.2	95.1	91.4
New object	40.1	-	-	-	80.7

Table 1: Performance of the individual and combined regrasping policies.

Evaluation of general regrasping policy After training individual policies we create a "transfer" dataset with grasp corrections obtained from the individual linear regrasping policies for all objects. For each set of tactile features, we query the respective previously-learned linear policy for the corresponding grasp correction. We take the input features for the individual policies from the unsuccessful grasps in the open-source BiGS dataset [14]. The grasps in BiGS were collected in an analogous experimental setup and can directly be used for creating the "transfer" dataset. In total, the training set contains 3351 examples: 1380 for the cylinder, 1035 for the box and 936 for the ball. We use supervised learning to learn a combined policy that mimics the behavior of the individual policies.

To find the optimal architecture of the neural network, we evaluated different networks with various depths and numbers of neurons to learn the nonlinear policy. The best performance is achieved by using 20 ST-HMP PCA features as inputs. We have not observed any improvement of the approximation accuracy when using more than one hidden layer. This indicates that the ST-HMP algorithm already extracts most of the distinctive features from the tactile data and we do not require additional deep network architecture for our task. The final neural network consists of one hidden layer of 20 hidden units with tangent sigmoid activation functions, 20 input features and 6 outputs for grasp position and orientation adjustments. The resulting number of parameters in the generalized policy is 546. Such a high-dimensional policy would be hard to learn by directly employing reinforcement learning. Our formulation as supervised learning, however, simplifies this problem and makes the learning process with relatively small amounts of data more feasible.

Table 1 shows performance of the generalized policy on the single objects. Interestingly, the combined policy achieves better performance on each of the single objects than the respective linear policies learned specifically for these object after one regrasp. Furthermore, in cases of the cylinder and the ball, the performance of the generalized policy is better than the linear policies evaluated after two regrasps. This shows that the general policy generalizes well between the single policies. In addition, by utilizing the knowledge obtained from single policies, the generalized policy performs better on the objects that the single policies were trained on.

The performance of the generalized policy on the box object is slightly worse than on the two other objects. A notable difference in this case is the increased importance of the gripper yaw angle with respect to the grasp performance. The individual policy learned on the box learns to correct the grasp such that the robot aligns its fingers with the box sides while regrasping. However, this is not important for the cylinder and the ball objects due to their symmetric shapes. Therefore, the regrasping policy for the box could not benefit from the two other policies when adjusting grasp in the yaw direction.

We test performance of the generalized policy on a novel, more complex object (see the bottom-right corner in Fig. 1), which was not present during learning. The generalized policy improves the grasping performance significantly, which shows its ability to generalize to more complex objects. Nevertheless, there are some difficulties when the robot performs regrasp on a part of the object that is different from the initial grasp. In this case, the regrasp is incorrect for the new part of the object, i.e. the yaw adjustment is suboptimal for the box part of the object due to the round grasping surface (the lid) in the initial grasp.

During the experiments, we were able to observe many intuitive corrections made by the robot using the learned regrasping policy. The robot was able to identify if one of the fingers was only barely touching the object's surface, causing the object to rotate in the hand. In this case, the regrasp resulted in either rotating or translating the gripper such that all of its fingers were firmly touching the object. Another noticeable trend learned through reinforcement learning was that the robot would regrasp the middle part of the object which was closer to the center of mass, hence, more stable for grasping. On the box object, the robot learned to change its grasp such that its fingers were aligned with the box's sides. These results indicate that not only can the robot learn a set of linear regrasping policies for individual objects, but also that it can use them as the basis for guiding the generalized regrasping behavior.

Conclusions

In this work, we proposed a method that is able to learn complex high-dimensional policies by using examples from simple policies learned through reinforcement learning. In this manner, we do not require large amounts of data to learn complex policies. Instead, we employed supervised learning techniques to mimic various behaviors of simple policies.

To show the effectiveness of our method, we applied it to the problem of regrasping using tactile features. In particular, we used early grasp stability prediction during the initial phases of the grasp and a regrasping behavior that corrects failed grasps based on tactile perception and improves over time.

Our experiments indicate that the combined policy learned using our method is able to achieve better performance on each of the single objects than the respective linear policies learned using reinforcement learning specifically for these objects after one regrasp. Moreover, the general policy achieves approximately 80% success rate after one regrasp on a novel object that was not present during training. These results show that our supervised policy learning method applied to regrasping can generalize to more complex objects.

As a next step, we plan to use the supervised policy learning method to learn other, more complex manipulation tasks. We also hope to be able to extend the presented method with other sensor modalities such as vision. Since the final regrasping policy is represented as a neural network, we believe that it is possible to combine other sensor modalities in a multi-modal deep learning setup by enforcing the networks for each of the modalities to share a subset of their parameters.

References

[1] J. Bohg, A. Morales, T. Asfour, and D. Kragic. Data-driven grasp synthesis a survey. *Robotics, IEEE Transactions on*, 30 (2):289–309, 2014.

[2] Y. Chebotar, K. Hausman, Z. Su, G.S. Sukhatme, and S Stefan. Self-supervised regrasping using spatio-temporal tactile features and reinforcement learning. In *IROS*, 2016.

[3] M.P. Deisenroth, G. Neumann, and J. Peters. A survey on policy search for robotics. *Foundations and Trends in Robotics*, 2(1-2):1–142, 2013.

[4] S. Levine, N. Wagener, and P. Abbeel. Learning contact-rich manipulation skills with guided policy search. In *ICRA*, 2015.

[5] C. Finn, X.Y. Tan, Y. Duan, T. Darrell, S. Levine, and P. Abbeel. Deep spatial autoencoders for visuomotor learning. *CoRR*, 117(117):240, 2015.

[6] M. Madry, L. Bo, D. Kragic, and D. Fox. St-hmp: Unsupervised spatio-temporal feature learning for tactile data. In *ICRA*, 2014.

[7] L. Bo, X. Ren, and D. Fox. Hierarchical matching pursuit for image classification: Architecture and fast algorithms. In *NIPS*, pages 2115–2123, 2011.

[8] M. Aharon, M. Elad, and A. Bruckstein. k -svd: An algorithm for designing overcomplete dictionaries for sparse representation. *Signal Processing, IEEE Transactions on*, 54 (11):4311–4322, 2006.

[9] I. T. Jolliffe. *Principal component analysis*. Springer, New York, 1986.

[10] J. Peters, K. Mülling, and Y. Altun. Relative entropy policy search. In *AAAI*. AAAI Press, 2010.

[11] S. Levine and V. Koltun. Guided policy search. In *ICML*, 2013.

[12] N. Wettels, V.J. Santos, R.S. Johansson, and G.E. Loeb. Biomimetic tactile sensor array. *Advanced Robotics*, 22(8): 829–849, 2008.

[13] Z. Su, K. Hausman, Y. Chebotar, A. Molchanov, G.E. Loeb, G.S. Sukhatme, and S. Schaal. Force estimation and slip detection/classification for grip control using a biomimetic tactile sensor. In *Humanoids*, 2015.

[14] Y. Chebotar, K. Hausman, Z. Su, A. Molchanov, O. Kroemer, G. Sukhatme, and S. Schaal. Bigs: Biotac grasp stability dataset. In *Grasping and Manipulation Datasets, ICRA 2016 Workshop on*, 2016. URL `http://bigs.robotics.usc.edu/`.