2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Daejeon Convention Center
October 9-14, 2016, Daejeon, Korea

Self-Supervised Regrasping using Spatio-Temporal Tactile Features and Reinforcement Learning

Yevgen Chebotar Karol Hausman Zhe Su Gaurav S. Sukhatme Stefan Schaal

***Abstract*— We introduce a framework for learning regrasping behaviors based on tactile data. First, we present a grasp stability predictor that uses spatio-temporal tactile features collected from the early-object-lifting phase to predict the grasp outcome with a high accuracy. Next, the trained predictor is used to supervise and provide feedback to a reinforcement learning algorithm that learns the required grasp adjustments based on tactile feedback. Our results gathered over more than 50 hours of real robot experiments indicate that the robot is able to predict the grasp outcome with 93% accuracy. In addition, the robot is able to improve the grasp success rate from 42% when randomly grasping an object to up to 97% when allowed to regrasp the object in case of a predicted failure.**

I. Introduction

Autonomous grasping of unknown objects is a fundamental requirement for service robots performing manipulation tasks in real world environments. Even though there has been a lot of progress in the area of grasping, it is still considered an open challenge and even the state-of-the-art grasping methods may result in failures. Two questions arise immediately: i) how to detect these failures early, and ii) how to adjust a grasp to avoid failure and improve stability.

Early grasp stability assessment is particularly important in the regrasping scenario, where, in case of predicted failure, the robot must be able to place the object down in the same position in order to regrasp it later. In many cases, early detection of grasping failures cannot be performed using a vision system as they occur at the contact points and involve various tactile events such as incipient slip. Recent developments of tactile sensors [1] and spatio-temporal classification algorithms [2] enable us to use informative tactile feedback to advance grasping-failure-detection methods.

In order to correct the grasp, the robot has to be able to process and use the information on why the grasp has failed. Tactile sensors are one source of this information. Our grasp adjustment approach aims to use this valuable tactile information in order to infer a local change of the gripper configuration that will improve the grasp stability. An example of a regrasping behavior is depicted in Fig. 1.

In this paper, we jointly address the problems of grasp-failure detection and grasp adjustment, i.e. *regrasping* using tactile feedback. In particular, we use a failure detection method to guide and self-supervise the regrasping behavior using reinforcement learning. In addition to the regrasping approach, we present an extensive evaluation of a spatio-temporal grasp stability predictor that is used on a biomimetic tactile sensor. This prediction is then used as a reward signal that supervises the regrasping reinforcement learning process.

Yevgen Chebotar, Karol Hausman, Gaurav S. Sukhatme and Stefan Schaal are with the Department of Computer Science; Zhe Su is with the Department of Biomedical Engineering, University of Southern California, Los Angeles. ychebota@usc.edu

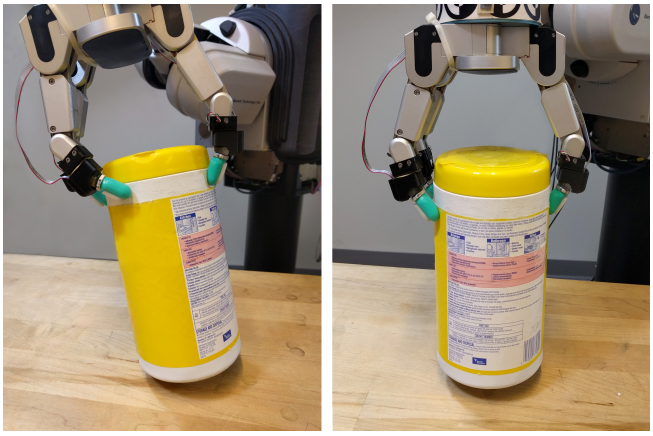

Fig. 1: *Regrasping scenario:* the robot partially misses the object with one of the fingers during the initial grasp (left), predicts that the current grasp will be unstable, places the object down, and adjusts the hand configuration to form a firm grasp of the object using all of its fingers (right).

The key contributions of our approach are: a) a regrasping framework that employs policy learning based on spatio-temporal tactile features, b) a high-accuracy grasp stability predictor for a biomimetic tactile sensor c) a self-supervised experimental setup that enables the robot to autonomously collect large amounts of data, d) an extensive evaluation of the grasp stability predictor and the learned regrasping behavior through more than 50 hours of real robot experiments.

II. Related Work

The problem of grasp stability assessment has been addressed previously in various ways. Dang and Allen [3] utilize tactile sensory data to estimate grasp stability. However, their approach does not take into account the temporal properties of the data and uses the tactile features only from one time step at the end of the grasping process. There have also been other approaches that model the entire time series of tactile data. In [4] and [5], the authors train a Hidden Markov Model to represent the sequence of tactile data and then use it for grasp stability assessment. The newest results in the analysis of the tactile time series data for grasp stability assessment were presented in [2]. The authors show that the unsupervised feature learning approach presented in their work achieves better results than any of the previously

978-1-5090-3762-9/16/$31.00 ©2016 IEEE

mentioned methods, including [4] and [5]. In this work, we show how the spatio-temporal tactile features developed in [2] can be applied to the state-of-the-art biomimetic tactile sensor, which enables us to better exploit the capabilities of this advanced tactile sensor.

Biomimetic tactile sensors and human-inspired algorithms have been used previously for grasping tasks. In [6], the authors show an approach to control grip force while grasping an object using the same biomimetic tactile sensor that is used in this work. Veiga et al. [7] use the same sensor to learn slip prediction classifiers and utilize them in the feedback loop of an object stabilization controller. Su et al. [8] present a grasping controller that uses estimation of various tactile properties to gently pick up different objects using a biomimetic tactile sensor. All of these works tackle the problem of human-inspired grasping. In this work, however, we focus on the regrasping behavior and grasp stability prediction. We also compare the grasp stability prediction results using the finger forces estimated by the method from [8] to the methods described in this work.

Reinforcement learning has enjoyed success in many different applications including manipulation tasks such as playing table tennis [9] or executing a pool stroke [10]. Reinforcement learning methods have also been applied to grasping tasks. Kroemer et al. [11] propose a hierarchical controller that determines where and how to grasp an unknown object. The authors use joint positions of the hand in order to determine the reward used for optimizing the policy. Another approach was presented by Montesano and Lopes [12], where the authors address the problem of actively learning good grasping points from visual features using reinforcement learning. In this work, we present a different approach to grasping using reinforcement learning, i.e. regrasping. In addition, we use tactile-feature-based reward function to improve the regrasping performance.

Tactile features have been rarely used in reinforcement learning manipulation scenarios. Pastor et al. [10] use pressure sensor arrays mounted on the robot's fingers to learn a manipulation skill of flipping a box using chopsticks. Similarly to [10], Chebotar et al. [13] use dynamic movement primitives and reinforcement learning to learn the task of gentle surface scraping with a spatula. In both of these cases, the robot was equipped with the tactile matrix arrays. In this work, we apply reinforcement learning to the task of regrasping using an advanced biomimetic tactile sensor together with state-of-the-art spatio-temporal feature descriptors.

The works most related to this paper have focused on the problem of grasp adjustment based on sensory data. Dang and Allen [3] tackles the regrasping problem by searching for the closest stable grasp in the database of all the previous grasps performed by the robot. A similar approach is presented by [14], where the authors propose a grasp adaptation strategy that searches a database for a similar tactile sensing experience in order to correct the grasp. The authors introduce an object-level impedance controller whose parameters are adapted based on the current grasp stability estimates. The grasp adaptation is focused on in-hand adjustments rather than placing the object down and regrasping it. The main differences between these approaches and ours are twofold: i) we employ spatio-temporal features and use tactile data from the beginning of the lifting phase which enables us to achieve high grasp stability prediction accuracy in a short time, and ii) we use reinforcement learning with spatio-temporal features which is supervised by the previously learned grasp stability predictor. This allows us to learn the regrasping behavior in an autonomous and efficient way.

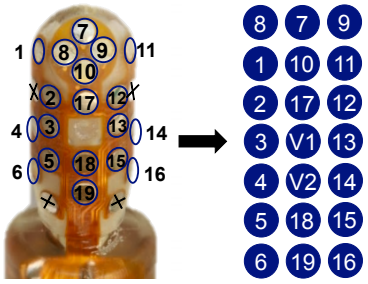

Fig. 2: The schematic of the electrode arrangements on the BioTac sensor (left). Tactile image used for the ST-HMP features (right). The X values are the reference electrodes. The 19 BioTac electrodes are measured relative to these 4 reference electrodes. V1 and V2 are created by taking an average response of the neighboring electrodes: V1 = avg(E17, E18, E12, E2, E13, E3) and V2 = avg(E17, E18, E15, E5, E13, E3).

III. Grasp Stability Estimation

The first component of our regrasping framework is the grasp stability predictor that provides early detection of grasp failures. We use short sequences of tactile data to extract necessary information about the grasp quality. This requires processing tactile data both in the spatial and the temporal domains. In this section, we first present the biomimetic tactile sensor used in this work and how its readings are adapted for the feature extraction method. To the best of our knowledge, this is the first time that the spatio-temporal tactile features are used for this advanced biomimetic tactile sensor. Next, we describe the spatio-temporal method for learning a sparse representation of the tactile sequence data and how it is used for the grasp stability prediction.

A. Biomimetic tactile sensor

In this work, we use a haptically-enabled robot equipped with 3 biomimetic tactile sensors - BioTacs [1]. Each BioTac consists of a rigid core housing an array of 19 electrodes surrounded by an elastic skin. The skin is inflated with an incompressible and conductive liquid. When the skin is in contact with an object, the liquid is displaced, resulting in distributed impedance changes in the electrode array on the surface of the rigid core. The impedance of each electrode tends to be dominated by the thickness of the liquid between the electrode and the immediately overlying skin. Since the BioTac uses liquid for its core functionality, it is a very

difficult sensor to simulate. To the best of our knowledge, there is no reliable simulation that is able to mimic the sensor's functionality.

In order to apply feature extraction methods from computer vision, as described in the next section, the BioTac electrode readings have to be represented as tactile images. To form such a 2D tactile image, the BioTac electrode values are laid out according to their spatial arrangement on the sensor as depicted in Fig. 2.

B. Hierarchical Matching Pursuit

To describe a time series of tactile data, we employ a spatio-temporal feature descriptor - Spatio-Temporal Hierarchical Matching Pursuit (ST-HMP). We choose ST-HMP features as they have been shown to have high performance in temporal tactile data classification tasks [2].

ST-HMP is based on Hierarchical Matching Pursuit (HMP), which is an unsupervised feature-extraction method used for images [15]. HMP creates a hierarchy of several layers of simple descriptors using a matching pursuit encoder and spatial pooling.

To encode data as sparse codes, HMP learns a dictionary of codewords using the common codebook-learning method K-SVD [16]. Given a set of N H-dimensional observations (e.g. image patches) $Y = [y_1, \ldots, y_N] \in R^{H \times N}$, HMP learns a M-word dictionary $D = [d_1, \ldots, d_M] \in R^{H \times M}$ and the corresponding sparse codes $X = [x_1, \ldots, x_N] \in R^{M \times N}$ that minimize the reconstruction error between the original and the encoded data:

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

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 optimization problem of minimizing the reconstruction error can be solved in an alternating manner. First, the dictionary D is fixed and the sparse codes are optimized:

$$\min_{x_i} \|y_i - Dx_i\|^2 \ \ s.t. \ \ \|x_i\|_0 \leq K.$$

Next, given the sparse code matrix X, the dictionary is recomputed by solving the following optimization problem for each codeword in the dictionary:

$$\min_{d_m} \|Y - DX\|_F^2 \ \ s.t. \ \ \|d_m\|_2 = 1.$$

The first step is combinatorial and NP-hard but there exist algorithms to solve it approximately. HMP uses orthogonal matching pursuit (OMP) [17] to approximate the sparse codes. OMP proceeds iteratively by selecting a codeword that is best correlated with the current residual of the reconstruction error. Thus, at each iteration, it selects the codewords that maximally reduce the error. After selecting a new codeword, observations are orthogonally projected into the space spanned by all the previously selected codewords. The residual is then recomputed and a new codeword is selected again. This process is repeated until the desired sparsity level K is reached.

The dictionary optimization step can be performed using a standard gradient descent algorithm.

C. Spatio-temporal HMP

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. As the next step, the sparse codes are aggregated using spatial max-pooling. In particular, the image is divided into spatial cells and each cell's features are computed by max-pooling all the sparse codes inside the spatial cell:

$$F(C_s) = \left[\max_{j \in C_s} |x_{j1}|, \ldots, \max_{j \in C_s} |x_{jm}| \right],$$

where $j \in C_s$ indicates that the image patch is inside the spatial cell C_s and m is the dimensionality of the sparse codes. The HMP features of each tactile image in the time sequence are computed by performing max-pooling on various levels of the spatial pyramid, i.e. for different sizes of the spatial cells.

After computing the HMP features for all tactile images in the time series, the pooling is performed on the temporal level by constructing a temporal pyramid. The tactile sequence is divided in sub-sequences of different lengths. For all sub-sequences, the algorithm performs max-pooling of the HMP features resulting in a single feature descriptor for each sub-sequence. 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: $F_P = [C_{11}, \ldots, C_{ST}]$, where S is the number of the spatial cells and T is the number of the temporal cells. Hence, the total dimensionality of the ST-HMP descriptor is $S \times T \times M$.

After extracting the ST-HMP feature descriptor from the tactile sequence, we use Support Vector Machine (SVM) with a linear kernel to learn a classifier for the grasp stability prediction as described in [2]. The tactile sequences used in this work consist of tactile data shortly before and after starting to pick up a grasped object. This process is described in more detail in Sec. V.

In this work, we also compare features extracted only from tactile sensors with combinations of tactile and non-tactile features, such as force-torque sensors, strain gages, finger angles, etc. To achieve temporal invariance, we apply temporal max-pooling to these features with the same temporal pyramid as for the tactile features. By doing so, we can combine tactile ST-HMP and non-tactile temporally pooled features by concatenating their feature vectors.

IV. Reinforcement learning for regrasping

Once a grasp is predicted as a failure 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. The parameters of this mapping function are learned using a reinforcement learning approach. In the following, we explain how this mapping is computed. In addition, we describe the policy search method and our approach to reducing the dimensionality of the problem. Note that the fact that we use tactile sensors to learn a policy necessitates learning how to regrasp rather than how to grasp directly.

A. Mapping from tactile features to grasp adjustments

Similarly to the grasp stability prediction, we use the ST-HMP features as the tactile data representation to compute the adjustment of the unsuccessful grasp. In particular, we use a linear combination of the ST-HMP features to compute the change of the grasp pose. The weights of this combination are learned using a policy search algorithm described in the next section.

Using multiple levels in the spatial and temporal pyramids of ST-HMP increases the dimensionality of tactile features substantially. This leads to a large number of parameters to learn for the mapping function, which is usually a hard task for policy search algorithms [18]. Therefore, we perform principal component analysis (PCA) [19] on the ST-HMP features and use only the largest principal components to compute the mapping.

The grasp pose adjustment is represented by the 3-dimensional change of the gripper's position and 3 Euler angles describing the change of the gripper's orientation. Each adjustment dimension is computed separately using the largest principal components of the ST-HMP features:

$$\begin{pmatrix} x \\ y \\ z \\ \alpha \\ \beta \\ \gamma \end{pmatrix} = \begin{pmatrix} w_{x,1} & \cdots & w_{x,n} \\ w_{y,1} & \cdots & w_{y,n} \\ \vdots & \ddots & \vdots \\ w_{\gamma,1} & \cdots & w_{\gamma,n} \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_n \end{pmatrix},$$

where $w_{i,j}$ are the weights of the tactile features ϕ_i and n is the number of principal components.

B. Policy search for learning mapping parameters

The linear combination weights of the mapping from the tactile features to the grasp adjustments $w_{i,j}$ are learned using reinforcement learning. For that, all the combination weights are concatenated to form a single parameter vector:

$$\boldsymbol{\theta} = (w_{x,1}, \ldots, w_{x,n}, \ldots, w_{\gamma,n}).$$

We define the policy $\pi(\boldsymbol{\theta})$ as a Gaussian distribution over $\boldsymbol{\theta}$ with a mean $\boldsymbol{\mu}$ and a covariance matrix $\boldsymbol{\Sigma}$. In order to find good regrasping candidates, the parameter vector $\boldsymbol{\theta}$ is sampled from this distribution. In the next step, we compute the reward $R(\boldsymbol{\theta})$ by estimating the success of the adjusted grasp using the grasp stability predictor described in Sec. III. After a number of trials is collected, the current policy is optimized and the process repeats. For policy optimization we use the relative entropy policy search (REPS) algorithm [20]. 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.

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. Using this notation, we can formulate a constrained optimization problem:

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

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

$$\int \pi(\boldsymbol{\theta})\, d\boldsymbol{\theta} = 1,$$

where, as mentioned before, $J(\pi)$ is the total expected reward of using the policy $\pi(\boldsymbol{\theta})$. The first constraint bounds the KL-divergence between the policies with the maximum information lost set to ϵ. The second constraint ensures that $\pi(\boldsymbol{\theta})$ forms a proper probability distribution.

Solving the optimization problem with Lagrange multipliers results in the following dual function:

$$g(\eta) = \eta\epsilon + \eta \log \int q(\boldsymbol{\theta}) \exp\left(\frac{R(\boldsymbol{\theta})}{\eta}\right) d\boldsymbol{\theta},$$

where the integral term can be approximated from the samples using the maximum-likelihood estimate of the expected value. Furthermore, from the Lagrangian it follows that:

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

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 $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N\}$ and corresponding episode rewards, the policy update rules for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ can be formulated as follows [18]:

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

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

Fig. 3: *Left:* Experimental setup used for learning the grasp stability predictor and the regrasping behavior. If the object falls out of the hand, it returns to its initial position due to the convex shape of the bowl. *Right:* The novel object (on the right) used for testing the regrasping behavior in comparison to the original object (on the left).

V. Evaluation and Discussion

In this section we describe the experimental setups and the results obtained for both parts of the regrasping framework: grasp stability prediction and the regrasping behavior learned using reinforcement learning.

A. Evaluation of grasp stability prediction

1) Experimental setup: We evaluate our method on the Barrett Arm manipulator and the Barrett hand equipped with three BioTac sensors. For the grasping experiments, we use a cylindrical object depicted in Fig. 3 on the left, whose position is known in advance. In addition, we introduce a bowl that is firmly attached to the table. The bowl is used to bring the object up right if it falls out of the gripper during the extensive shaking motions that are performed later in the experiment. This modification enables us to fully automate the learning process and let the robot run for more than 20 hours to autonomously learn the grasp stability predictor.

The experiment proceeds as follows. The robot reaches for the object to perform a top grasp, which we refer to as a *nominal grasp*. In order to achieve more variety in the data, we generate subsequent grasps by adding white noise to the nominal grasp. We refer to these grasps as *random grasps*. The std. dev. of the white noise is $\pm 10deg$ in roll and pitch of the gripper, $\pm 60deg$ in yaw, and $\pm 1cm$ in all translational directions. These parameters are tuned such that there is always at least one finger touching the object. After approaching and grasping the object using the force grip controller described in [8], the robot picks the object up. To ensure that the grasp is stable the robot performs a range of shaking motions by rapidly changing the end-effector orientation by $\pm 15deg$ and position by $\pm 3cm$ in all directions multiple times. If the object is still in the hand after the shaking motions, we consider it to be a successful grasp. The wrist-mounted force-torque sensor is used to determine if the object is still in the hand at the end of the experiment. We collected 1000 grasps, out of which 46% resulted in failures and 54% succeeded. The data collected throughout the grasp stability prediction experiments are available online[1] and described in detail in [21].

Features	# of grasps	Avg. Accuracy, %
Electrodes	500	90.73
Finger angles	500	80.55
Strain gages	500	84.91
Hand orientation	500	74.00
Force-torque	500	69.63
BioTac finger forces	500	88.55
All features	500	91.09
All features	1000	93.00

TABLE I: Results obtained for various sensor modalities for the grasp stability predictor using ST-HMP. The combination of all modalities outperforms any other single modality.

To extract tactile features, we use a temporal window of $650ms$ before and $650ms$ after starting picking up the object. Our goal is to determine if the grasp is going to fail early in the picking up phase. This way, we can stop the motion early enough to avoid displacing the object, and thus, enable the robot to perform regrasping. In this work, we concatenate the tactile images from all three BioTacs as described in [2] to form a single image for ST-HMP feature extraction. HMP learns a dictionary of size $M = 100$ with the sparsity level set to $K = 4$. The spatial pooling is performed with a 3 level pyramid: the image is divided in 1x1, 2x2 and 3x3 cell grids, which results in $S = (1 + 2^2 + 3^2) = 14$ spatial cells. The temporal pyramid consists of 5 max-pooling levels: the sequence is divided into 1, 2, 4, 8 and 16 parts. This results in $T = (1+2+4+8+16) = 31$ temporal cells. In addition, in order to take into account the signs of HMP features, which are lost due to max-pooling on absolute values, we save the feature vector elements with both positive and negative signs as described in [22]. This doubles the size of the feature descriptor. Hence, the total amount of the ST-HMP features is $S \times T \times M \times 2 = 14 \times 31 \times 100 \times 2 = 86800$.

2) Results: To evaluate the grasp stability prediction, we perform a 5-fold cross-validation on our data set by using 20% of the data as the test set. We compare the prediction results using different ST-HMP features constructed from different sensor modalities available on the robot: electrode values in the BioTac sensors, joint position of the fingers, strain gages that are mounted in the fingers, the orientation of the hand, wrist-mounted force-torque sensor values and the finger forces computed from the BioTac electrodes data using the algorithm presented in [8].

Table I shows the average classification accuracy values for different sensory modalities trained on 500 grasps. ST-HMP features extracted from the raw electrode values of the BioTac sensors gives a better performance than any other modality alone. Combining different modalities leads to only a slight accuracy increase, from 90.73% to 91.09%, which indicates that most of the grasp outcome information is already contained in the electrode values. Interestingly, the performance of using the forces estimated from the BioTacs is comparable to using the raw BioTac electrode values (88.55% and 90.73%, respectively). This indicates that finger

[1] http://bigs.robotics.usc.edu/

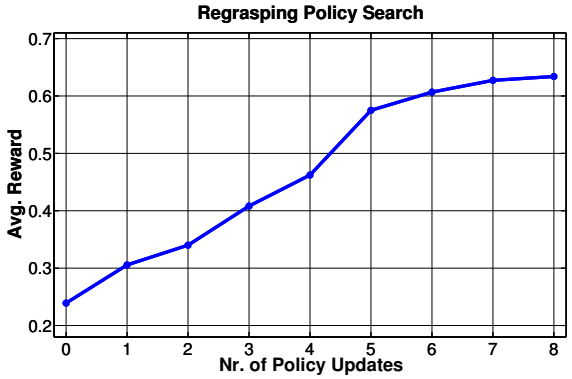

Fig. 4: Reinforcement learning curve for regrasping using REPS. Policy updates are performed after each 100 regrasps. The average reward increases after each policy update. The policy converges after 8 updates.

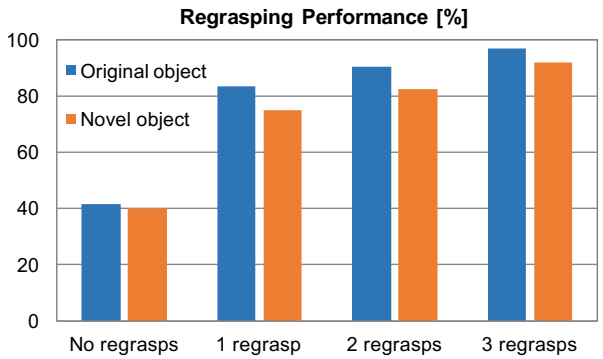

Fig. 5: Results for the regrasping experiment using the learned policy. Evaluated over 100 random grasps on the original object used for training the policy (blue) and a novel object (orange). There is a significant increase of the grasp success rate with additional regrasps.

contact forces already contain a large amount of information about the grasp success and HMP with dictionary learning is able to implicitly extract this information. However, the better performance of the grasp stability prediction using the raw BioTac electrode values indicates that there are more clues about the grasp success that can be extracted from the electrode values than contained in the estimated force values.

In order to boost the performance of the grasp stability predictor we additionally increased the training dataset from 500 to 1000 grasps. This further improves the accuracy of the method that fuses all the modalities from from 91.09% to 93.00%. This indicates that with larger amounts of data, the learning method used in this work is able to better generalize to the problem of grasp stability prediction.

B. Evaluation of the regrasping behavior

1) Experimental Setup: After learning the grasp stability predictor, we evaluate the regrasping algorithm, which uses the feedback provided by the stability prediction. The experimental setup is similar to the one for the grasp stability predictor. Since the robot can self-supervise by using the stability prediction, we are able to let the robot run for more than 30 hours to autonomously learn the regrasping behavior.

The dimensionality of ST-HMP features becomes large with the growing number of spatio-temporal cells. As mentioned earlier, the dimensionality of our feature descriptor is 86800. As described in Sec. IV-A, we apply PCA and extract 5 principal components from the grasp stability training data collected over 1000 grasps. Our policy contains 30 parameters (5 for each of the 6 grasp adjustment dimensions).

To evaluate the policy search, we collect 9 sets of 100 regrasping samples using the currently learned policy, resulting in a total of 900 policy samples. The policy is updated after each set. The regrasping procedure is as follows: first, we perform a random grasp in the same way as during grasp stability data collection. If it is unsuccessful, the algorithm samples the current policy and the robot performs up to 3 regrasps. If one of the regrasps is successful, the robot stops regrasping and performs the next random grasp. The following rewards are saved for all policy samples:

0.0 - The grasp is predicted unsuccessful by the stability predictor. We do not perform any additional actions.
0.5 - The grasp is predicted successful by the stability predictor, which, due to its high accuracy, corresponds to a stable grasp at the beginning. However, the object falls out of the hand after additional extensive shaking motions.
1.0 - The grasp is predicted successful and the object is still in the hand after additional shaking motions.

2) Results: Fig. 4 shows the average reward values after each policy update. The robot is able to improve its regrasping behavior significantly. The policy starts at the average reward of 0.24 and converges to 0.63 after 8 policy updates. This means that the average regrasp achieved with the learned policy was almost always stable when picking up the object and very often it was stable enough to stay in hand during the extensive shaking motions. During the experiments, we were able to see many intuitive corrections made by the robot by using the tactile feedback. 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 hand. In this case, the generated 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 central part of the object which was closer to the center of mass, hence more stable for grasping. Moreover, the texture of the lid of the object used in the experiments was very smooth which often resulted in the object sliding down. However, the surface below the lid was rougher, which leads to a better grasp, and the robot was able to find this subtlety by using the tactile feedback.

In order to evaluate the final policy learned by our approach, we perform additional 100 random grasps of the object that we used for learning the policy and another 100 random grasps of a novel object (Fig. 3 right). The new object is both smaller and lighter. However, it still has a similar cylindrical shape. The robot has 3 attempts to regrasp each of the objects using the learned policy. Fig. 5 shows the percentage of successful grasps using the initial random

grasp and allowing the robot to perform 1, 2 or 3 regrasps. In this case, the grasp is classified as successful if the object is still in hand after the shaking motions. Already after 1 regrasp, the robot is able to correct the majority of the failed grasps. The success rate achieved with the original object on which the policy was learned increases from 41.8% to 83.5%. Moreover, allowing additional regrasps increases this value to 90.3% for 2 and 97.1% for 3 regrasps.

This shows that sampling the policy several times increases the probability of a successful regrasp. A likely explanation for this is that the policy variance stays high due to existence of high-reward samples that lie far from each other. To exactly represent such a distribution, a more complex policy is needed, which is the area of future work.

The regrasping success rate on the novel object is slightly lower than on the original one. Since the novel object is smaller, the policy tends to overshoot the pose correction, which was learned for the larger object. However, we are still able to achieve a significant improvement of the grasping performance, resulting in an improvement from 40.2% to 75.2% success rate after 1 regrasp, 82.5% and 92.1% after 2 and 3 regrasps respectively. These results indicate that the robot is able to learn a regrasping strategy that generalizes to a novel object using a policy learned on another object of a similar shape but different size and physical properties.

VI. Conclusions and Future Work

In this work, we presented a framework for learning regrasping behaviors based on tactile data. We trained a grasp stability predictor that uses spatio-temporal tactile features collected from the early-object-lifting phase to predict the grasp outcome with high accuracy. The trained predictor was used to supervise and provide feedback to a reinforcement learning algorithm that also uses spatio-temporal features extracted from a biomimetic tactile sensor to estimate the required grasp adjustments. In order to test our approach, we collected over 50 hours of evaluation data on a real robot. We were able to achieve a 93% grasp prediction accuracy and improve grasp success rate of a single object from 41.8% to 97.1% (and on a novel object from 40.2% to 92.1%) by allowing up to 3 regrasps of the object.

In the future work, we plan to use a more complex policy class to better cover the distribution of the correct grasp adjustments and reduce the variance of the policy samples. Additionally, we plan to evaluate our approach on a larger amount of objects and learn regrasping behaviors that work across a wide range of different objects and scenarios.

References

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

[2] M. Madry, L. Bo, D. Kragic, and D. Fox. St-hmp: Unsupervised spatio-temporal feature learning for tactile data. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 2262–2269, May 2014.

[3] H. Dang and P.K. Allen. Stable grasping under pose uncertainty using tactile feedback. *Autonomous Robots*, 36(4):309–330, 2014.

[4] Y. Bekiroglu, J. Laaksonen, J.A. Jørgensen, V. Kyrki, and D. Kragic. Assessing grasp stability based on learning and haptic data. *Robotics, IEEE Transactions on*, 27(3):616–629, 2011.

[5] Y. Bekiroglu, D. Kragic, and V. Kyrki. Learning grasp stability based on tactile data and hmms. In *RO-MAN, 2010 IEEE*, pages 132–137. IEEE, 2010.

[6] N. Wettels, A.R. Parnandi, J. Moon, G.E. Loeb, and G.S. Sukhatme. Grip control using biomimetic tactile sensing systems. *Mechatronics, IEEE/ASME Transactions On*, 14(6): 718–723, 2009.

[7] F. Veiga, H. Van Hoof, J. Peters, and T. Hermans. Stabilizing novel objects by learning to predict tactile slip. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 5065–5072. IEEE, 2015.

[8] 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 *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*, pages 297–303, 2015.

[9] J. Kober, E. Oztop, and J. Peters. Reinforcement learning to adjust robot movements to new situations. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, page 2650, 2011.

[10] P. Pastor, M. Kalakrishnan, S. Chitta, E. Theodorou, and S. Schaal. Skill learning and task outcome prediction for manipulation. *IEEE International Conference on Robotics and Automation*, pages 3828–3834, 2011.

[11] O. Kroemer, R. Detry, J. Piater, and J. Peters. Combining active learning and reactive control for robot grasping. *Robotics and Autonomous Systems (RAS)*, 58(9):1105–1116, 2010.

[12] L. Montesano and M. Lopes. Active learning of visual descriptors for grasping using non-parametric smoothed beta distributions. *Robotics and Autonomous Systems*, 60(3):452–462, 2012.

[13] Y. Chebotar, O. Kroemer, and J. Peters. Learning robot tactile sensing for object manipulation. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 3368–3375. IEEE, 2014.

[14] M. Li, Y. Bekiroglu, D. Kragic, and A. Billard. Learning of grasp adaptation through experience and tactile sensing. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 3339–3346. IEEE, 2014.

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

[16] 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.

[17] Y.C. Pati, R. Rezaiifar, and P.S. Krishnaprasad. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In *Signals, Systems and Computers, Asilomar Conference on*, pages 40–44 vol.1, 1993.

[18] 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.

[19] I. T. Jolliffe. *Principal component analysis*. Springer, 1986.

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

[21] Y. Chebotar, K. Hausman, Z. Su, A. Molchanov, O. Kroemer, G.S. Sukhatme, and S. Schaal. Bigs: Biotac grasp stability dataset. In *ICRA 2016 Workshop on Grasping and Manipulation Datasets*, Stockholm, Sweden, May 2016.

[22] L. Bo, X. Ren, and D. Fox. Multipath sparse coding using hierarchical matching pursuit. In *Computer Vision and Pattern Recognition (CVPR)*, pages 660–667, 2013.