Template-Based Learning of Grasp Selection

Alexander Herzog, Peter Pastor, Mrinal Kalakrishnan, Ludovic Righetti, Tamim Asfour, Stefan Schaal

***Abstract*— The ability to grasp unknown objects is an important skill for personal robots, which has been addressed by many present and past research projects, but still remains an open problem. A crucial aspect of grasping is choosing an appropriate grasp configuration, i.e. the 6d pose of the hand relative to the object and its finger configuration. Finding feasible grasp configurations for novel objects, however, is challenging because of the huge variety in shape and size of these objects. Moreover, possible configurations also depend on the specific kinematics of the robotic arm and hand in use. In this paper, we introduce a new grasp selection algorithm able to find object grasp poses based on previously demonstrated grasps. Assuming that objects with similar shapes can be grasped in a similar way, we associate to each demonstrated grasp a grasp template. The template is a local shape descriptor for a possible grasp pose and is constructed using 3d information from depth sensors. For each new object to grasp, the algorithm then finds the best grasp candidate in the library of templates. The grasp selection is also able to improve over time using the information of previous grasp attempts to adapt the ranking of the templates. We tested the algorithm on two different platforms, the Willow Garage PR2 and the Barrett WAM arm which have very different hands. Our results show that the algorithm is able to find good grasp configurations for a large set of objects from a relatively small set of demonstrations, and does indeed improve its performance over time.**

I. INTRODUCTION

Autonomous robotic grasping is one of the pre-requisites for personal robots to become useful when assisting humans in daily life. Seemingly easy for humans, it still remains a very challenging task for robots. An essential aspect of robotic grasping is to automatically choose an appropriate grasp configuration given an object as perceived by the sensors of the robot. The high variety in the size and geometry of objects to be grasped (for example, household objects, see Fig. 5) makes it very hard to develop an algorithm that provides promising grasp hypotheses. Grasp planners have been proposed that require exact object models describing their size and geometry. These approaches usually search among all feasible grasp configurations and choose the one that maximizes a grasp quality metric. In [1] a large scale data base of object models is created such that partial sensor information of unknown objects can be matched in order to choose from pre-computed gripper-configurations. In [2] a set of object primitives considering shape and material properties together with stable grasps computed in GraspIt! [3] are used for generalization to other object models using a nearest neighbor metric. The algorithm, proposed in [4], computes the medial axes on point clouds sampled from object models and applies a set of heuristics to compute grasp configurations.

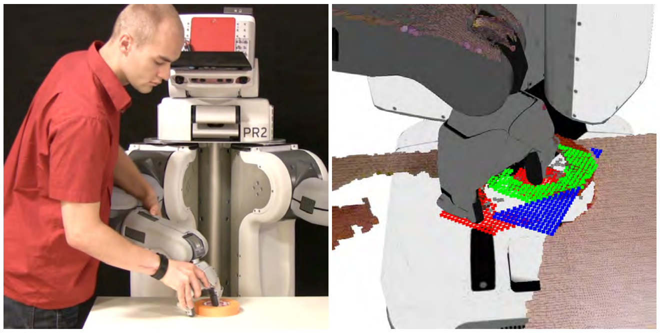

Fig. 1. User demonstrates a feasible grasp to the PR2 robot (left). Extracted template and gripper-pose (right).

However, online generation of object models, especially for articulated objects, remains challenging. Model free approaches have been proposed that directly operate on point clouds provided by 3d sensors (for example a stereo camera system, or the Microsoft Kinect). Hsiao et. al. developed an algorithm that searches among feasible top and side grasps to maximize the amount of object mass between the finger tips of the robot gripper [5]. Klingbeil et. al. developed an algorithm that searches for a good grasp configuration by maximizing the contact area between the robot's gripper and the perceived point cloud [6]. Both of these approaches generate a ranked list of grasp hypotheses suitable for execution on the robot. The ranking of these grasp hypotheses is fixed and does not adapt over time. Furthermore, these algorithms do not allow to add an appropriate grasp hypothesis if none of the generated grasp configurations lead to a successful grasp. In [7] and [8] models of objects with grasp densities are learned in form of hierarchies of features using *early cognitive vision* descriptors. In [9] grasp affordances are learned by trial-and-error from local features extracted from 2d images. In these algorithms grasps are executed for a fixed finger configuration such that only objects suited for the particular configuration can be grasped. The algorithms, presented in [10] and [11] use supervised learning with a set of features extracted from 2d and 3d vision to learn grasp configurations. However, the high amount of training data forces these algorithms to learn from simulated data rather than from executions in real world environments. In

The work described in this paper was partially conducted within the EU Cognitive Systems project GRASP (IST-FP7-IP-215821) funded by the European Commission and the German Humanoid Research project SFB588 funded by the German Research Foundation (DFG: Deutsche Forschungsgemeinschaft). Alexander Herzog received support from the InterACT - International Center for Advanced Communication Technologies.

This research was supported in part by National Science Foundation grants ECS-0326095, IIS-0535282, IIS-1017134, CNS-0619937, IIS-0917318, CBET-0922784, EECS-0926052, CNS-0960061, the DARPA program on Advanced Robotic Manipulation, the Army Research Office, the Okawa Foundation, and the ATR Computational Neuroscience Laboratories.

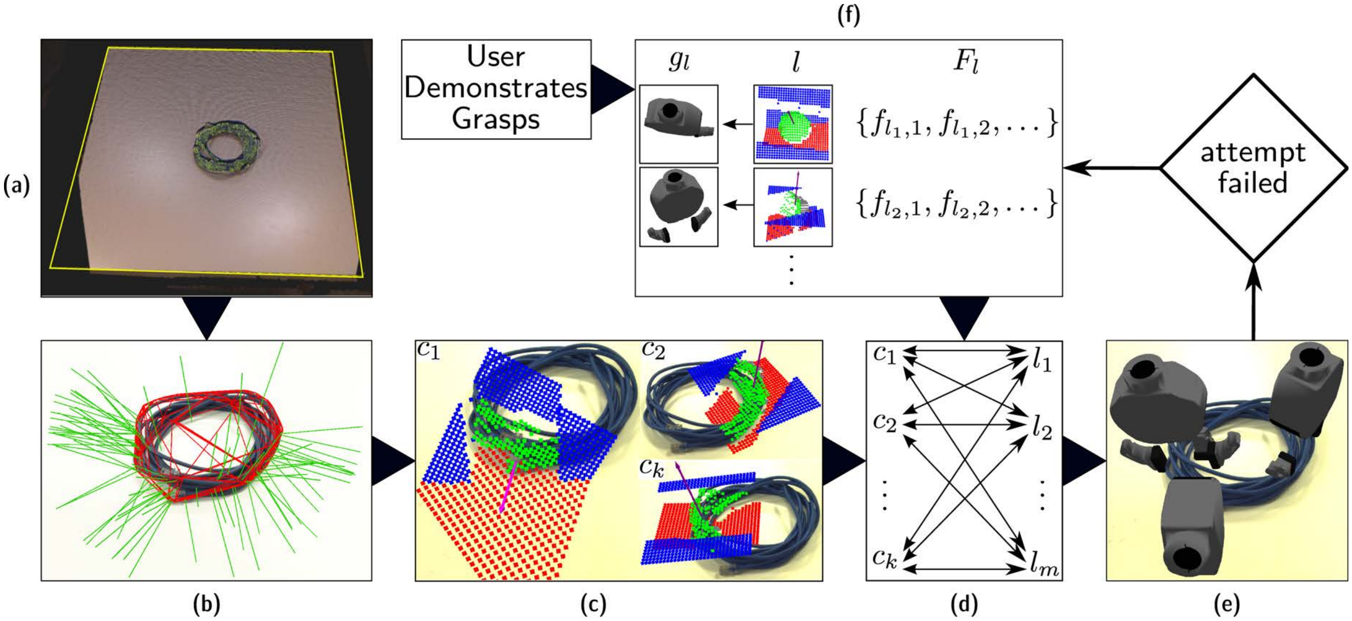

Fig. 2. Overview of the presented algorithm. (a) A 3d point cloud showing the object with background as it is perceived by the robot. The object (yellow point cloud) is segmented out from the table (yellow rectangle) for further processing. (b) The convex hull of the object (red) and its normals (green) are used as approximation of the object surface to reduce search space of templates. (c) Three of k template candidates extracted from the object relative to various (purple) height-axes. (d) Candidates are matched against a grasp library to lookup good grasp configurations. (e) The matching distance is used to rank resulting configurations. If a grasp attempt fails, feedback is returned to the grasp library and used in successive matching steps to adapt ranking. (f) New grasps are added by user demonstrations to the grasp library. Each template l is stored together with the according gripper configuration g_l and a set F_l of templates from failed grasp attempts.

this paper, we propose a novel model free grasp selection algorithm with the following favorable characteristics:

- Appropriate grasp poses together with finger configurations can be taught through kinesthetic teaching.
- Our proposed local shape descriptor, the template, encodes hand-sized regions on the object that are suitable for grasping such that it generalizes across different objects.
- Our algorithm is able to autonomously improve the ranking of generated grasp candidates over time based on feedback from previous grasp executions on real robots.
- The presented approach is applicable to different hand architectures.
- Finally, our proposed method is computationally efficient.

Our approach is based on the simple and common assumption that similarly shaped objects can be grasped with similar grasp configurations. Although, there might be more factors that influence the choice of a good grasp, (e.g. its surface or inertia properties) we will show that local shape patches provide a major feature for successful grasp selection. For example, a pen can be grasped from the table with a strategy similar to that used to grasp a screwdriver of the same size. Recently, templates have been successfully used to encode local regions of terrains enabling a quadruped robot to choose good footholds [12]. In [12], templates have been used to encode terrain heightmaps. In contrast, in our work, we use templates to encode object heightmaps that are sampled from various height-axes. We store known object shapes represented by templates together with feasible grasp configurations in a library. To obtain good grasp hypotheses for novel objects, our algorithm samples and compares shape patches to those patches in the library and retrieves the associated grasp configuration for the best match. An initial set of object shapes can be acquired by demonstrating feasible grasp configurations for a particular set of objects to the robot and store them as a template associated to a grasp pose. A grasp configuration is given by a 6 degrees-of-freedom (DOF) end effector pose as well as the joint configuration of the robot's gripper.

II. OVERVIEW

An overview of the proposed grasp selection algorithm is shown in Fig. 2. In Section II-A we describe the proposed templates in more detail. The heuristic used to sample templates from an object is described in Section II-B. In the following Section II-C we explain how new grasps can be added to the grasp library. Matching of sampled templates to the grasp library is described in Section II-D. In Section II-E we describe how feedback is used to improve ranking of grasp hypotheses.

A. Templates

We make the assumption that the region of an object being in contact with a gripper is most important for the success of a grasp attempt. One could think of a further step that takes holistic features as e.g. inertia of the object into account to improve ranking of the grasp hypotheses computed by our algorithm. However, a local descriptor has the advantage, that similar regions can be found on differently shaped objects

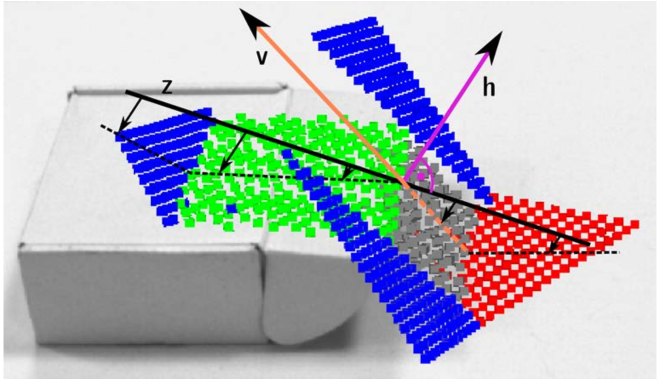

Fig. 3. One example template extracted from the point cloud of a carton box. Height values (small black arrows) are measured relative to a plane perpendicular to the height-axis h, here indicated by the black line z. Tiles of type *surface* (green) are extracted from the object surface. Tiles of type *background* (red) describe the height between z and the table. *Void* regions (blue) are bounded to the limits of the bounding box of the gripper. Tiles of type *occlusion* store the height of the upper bound of an occluded region. They depend on the viewpoint v and the detected object *surface*.

to ensure better generalization to new objects as seen e.g. in Fig. 4. We use templates to encode object heightmaps that are sampled from various height-axes as described in Section II-B. Besides a height-axis (purple arrows in Fig. 2c) the coordinate frame of a template is also defined by a rotation about the height-axis. The map is rasterized to $n \times n$ tiles, which encode the described object region by height values relative to the height-axis. In addition to a height value each tile also stores a region type. We distinguish four different types:

- Regions on the object *surface* are significant for the shape of the grasped object part.
- For a grasp configuration it is desirable to avoid premature contacts with the object. Thus fingers need to be fit in *void* regions.
- The *background*, i.e. the table needs to be treated differently, too. Finger should neither fit into those regions, nor they should enclose it.
- Regions that cannot be determined as one of the previous types due to *occlusion* are encoded as well.

Hence each tile containing a height value and a type is defined as

$$t \in T = \mathbb{R} \times \{surface,\ void,\ occlusion,\ background\}.$$

A template is defined as a vector of tiles $\mathbf{c} \in T^{n^2}$ with raster granularity n and template size set according to the gripper in use. As we are interested in the region grabbed by the gripper, a gripper pose $\mathbf{g_c} \in \mathbb{R}^6$ is associated to each template. It is either taught by a user or looked up in the grasp library in the matching step. The bounding box defined by $\mathbf{g_c}$ and the size of the gripper is used to limit height values to the the bounding box. The type of exceeding values is set to *void*. Templates are extracted from perceived point clouds as illustrated in Fig. 3.

B. Template Acquisition

Before height values can be extracted, it is necessary to define the template's coordinate frame relative to the object, i.e. a height-axis and a rotation about it. To make the algorithm computationally efficient, we limit the search space consisting of these coordinate frames. Therefore we sample height-axes that are perpendicular to the object surface. A rough approximation of the surface is obtained by computing the convex hull of the object point cloud. For each polygon of the resulting mesh we use the center and normal-vector to define a height-axis as shown in Fig. 2b. The rotations about one height-axis are discretized into r steps resulting in r templates per polygon. Hence the number of sampled templates depends on the number of polygons of the convex hull times r. We ignore polygons with normals pointing away from the viewpoint. As they are extracted from a convex hull, they are on the backside of the object (relative to the viewpoint). Approaching the object from such a direction results in infeasible grasps anyways.

C. Learning good grasp configurations from demonstration

An initial set of grasp configurations can be learned from demonstration through kinesthetic teaching. To add a good grasp configuration to the library the user is required to move the robot's gripper to a favorable grasp configuration as shown in Fig. 1. The proposed method then automatically extracts the template and stores the demonstrated finger configuration and gripper pose relative to this template into the library. Thus, each library entry consists of an extracted template $\mathbf{l} \in T^{n^2}$, an associated gripper pose $\mathbf{g_l} \in \mathbb{R}^6$ and a finger joint configuration. We want to emphasize that for extending the robot's grasp repertoire these grasp configurations can be taught by a user who is not required to have any expert knowledge.

D. Matching

This section describes how candidate templates $\mathbf{c} \in T^{n^2}$ are matched against library templates $\mathbf{l}$ to find similar object shapes known from demonstration. For this purpose we define the difference between templates as a weighted ℓ_1 distance. Both, geometrical shape of templates encoded by height values c_i, l_i and information from region types $\hat{c}_i, \hat{l}_i$ are combined to

$$\sigma(\mathbf{c}, \mathbf{l}) = o_{\mathbf{c},\mathbf{l}} \sum_{i}^{n^2} W_{\hat{c}_i, \hat{l}_i} \left| c_i - l_i \right|, \tag{1}$$

where $W \in \mathbb{R}^{4 \times 4}$ weights height distances according to region types. The lower σ is, the more similar are $\mathbf{c}$ and $\mathbf{l}$ to each other according to geometrical shape and types of region. W is a matrix of weights

	surface	*void*	*occlusion*	*background*
surface	50	50	50	50
void	50	12	12	12
occlusion	50	12	12	12
background	50	12	12	12

We use only two weights for the 16 possible combinations. A higher weight is used to give more importance to differences in heights of tiles that contain an object. Additionally the sum is weighted by a global normalization factor $o_{\mathbf{c},\mathbf{l}} \in \mathbb{R}$.

It is used to make templates with different number of surface tiles comparable. The normalization factor is defined as

$$o_{\mathbf{c},\mathbf{l}} = \frac{max\{\mathrm{surf}_{\mathbf{c}}, \mathrm{surf}_{\mathbf{l}}\}}{\mathrm{surf}_{\mathbf{c},\mathbf{l}}}$$

where $\mathrm{surf}_{\mathbf{c}}$, $\mathrm{surf}_{\mathbf{l}}$ are the number of *surface* tiles in **c**, **l** and $\mathrm{surf}_{\mathbf{c},\mathbf{l}}$ is the number of elements in $\{\hat{c}_i = \hat{l}_i \mid \hat{c}_i = \hat{l}_i = surface,\ i \in 1 \dots n^2\}$, which is the number of indices that refer to tiles that are of type *surface* in both templates. The objective is to maximize overlay of surface regions relative to their size.

E. Improving grasp selection from experience

So far the presented algorithm is capable of producing grasp hypotheses by sampling candidate templates from an unknown object and matching them against a library of known templates. The distance σ can further be used to rank grasp hypotheses among each other. The lower the distance, the higher the rank. However, we do not use σ directly, but define a matching function, that uses σ to improve matching over time.

The core assumption we use for our algorithm is that if **c** and **l** are similar, and a grasp $\mathbf{g_l}$ works for **l** we infer that $\mathbf{g_l}$ also works for **c**. In our approach we try resulting grasp hypotheses $\mathbf{g_l}$ on **c** to verify the inferred statement. In case $\mathbf{g_l}$ does not work for **c**, this implies that **c** and **l** were not similar in the first place. This allows to improve similarity matching between templates. However, the implication cannot be inverted. E.g. let **c** describe the top of a cup and let **l** describe the bottom of a cup standing upright on the table. The same overhead grasp can be applied in both cases, although the shapes are clearly not similar. Hence only negative grasp trials can be used to improve the similarity matching. Nevertheless, it is still possible to add successful grasps as new entries to the library. However, we do not consider this possibility in this contribution.

To consider templates from failed grasp attempts we extend the grasp-library by adding a set $F_{\mathbf{l}} \subset T^{n^2}$ of failures to each library template **l**. If a candidate template was matched to **l** and the related gripper-pose $\mathbf{g_l}$ led to a failed grasp trial, we add the candidate template to $F_{\mathbf{l}}$. Feedback is not applied globally to all library templates as it is related to a particular gripper-pose $\mathbf{g_l}$.

The presented matching function takes experience from failures into account and is composed from three template distances

$$\alpha = \sigma(\mathbf{c},\mathbf{l}),$$
$$\beta = min\{\sigma(\mathbf{c},\mathbf{f_i}) \mid \mathbf{f_i} \in F_{\mathbf{l}}\},$$
$$\gamma = min\{\sigma(\mathbf{l},\mathbf{f_i}) \mid \mathbf{f_i} \in F_{\mathbf{l}}\}.$$

If $F_{\mathbf{l}} = \emptyset$, $\beta = \gamma \simeq \infty$ (initial state of the grasp-library after demonstration). Additionally to the distance to a library template α we take the distance to the least different failure template β into account. γ is a quality measure for **l**. The higher it is, the more **l** is robust to variation in shape. We combine these three distances to a matching function

$$m(\mathbf{c},\mathbf{l}) = \frac{\alpha}{[1-exp(-3\beta^2)][1-exp(-10\gamma^2)]}, \qquad (2)$$

which yields a mismatch measurement for a candidate template $\mathbf{c} \in T^{n^2}$ and a library template $\mathbf{l} \in T^{n^2}$. The sigmoid-shaped functions in the nominator have the effect that **c** and **l**, if they are less different from failures, will receive a higher mismatch value. For each candidate the best matching library template is chosen. Additionally $m(\mathbf{c},\mathbf{l})$ is used to rank candidates among each other. The resulting gripper-poses are looked up in the grasp library and set relative to the coordinate frames of candidate templates. The highest ranked feasible grasp is applied to the unknown object.

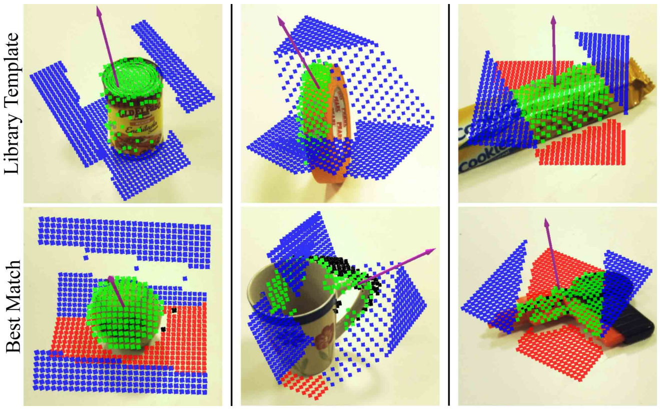

Fig. 4. The bottom row shows candidate templates from unknown objects; the top row shows the corresponding best match to the templates contained in the library. The purple arrows show the height-axes of the templates.

III. RESULTS

A. Experimental setup

We evaluated the presented grasp selection algorithm on two different robots with different end effectors.

1) PR2: We used the Willow Garage PR2[1] together with a head-mounted Microsoft Kinect for 3d point cloud extraction (see Fig. 1). Our algorithm was integrated in the pr2_tabletop_manipulation_pipeline. The gripper of the PR2 consists of two fingers and 1 DOF for closure (see Fig. 7). We used position control in order to move the gripper to the grasp pose and closed the fingers until the joint reached a globally predefined effort.

2) Barrett_Hand BH280: We further evaluated our approach on a Barrett WAM arm with a Barrett_Hand BH280[2] as end effector. A head-mounted and calibrated Asus Xtion sensor has been used to obtain point clouds. The grasp planner was used in combination with the motion planner, presented in [13]. The gripper pose was adapted by force control using the 6-axis force torque sensor in the wrist of the hand. We set a desired force of -1N along the approaching direction to guarantee that the palm is in contact with the object. A torque of 0Nm for pitch and yaw is applied to adapt for slightly miss-oriented grasp poses which lead to premature contact with the table. The other three dimensions

[1] http://www.willowgarage.com/pages/pr2/overview

[2] http://thearmrobot.com/aboutRobot.html

are not force controlled. These settings were set globally for all the experiments. Different to the PR2 the Barrett_Hand has three fingers with 1 DOF each. An additional DOF controls the spread between the left and right fingers resulting in 4 DOF (see Fig. 8). Although the hand architectures are quite different, the only parameter that had to be adapted was the size of templates, because it was set according to the gripper size. It was set to $15cm \times 15cm$ for the PR2 and to $25cm \times 25cm$ for the Barrett_Hand.

During the experiments, objects were placed in front of the robot on a table one at each time. A ranked list of grasp hypotheses was created using the grasp selection algorithm presented in Section II. The best feasible one was chosen by the planning environment and applied to the test object.

B. *Experiments*

1) Grasp selection from demonstrations: For the first experiment we demonstrated 15 grasps on a training set of 7 objects to the PR2 as described in Section II-C. The trained algorithm was tested on a set of 38 differently shaped objects (see Fig. 5). They were placed in different orientations to cover distinct viewpoints on the object. After grasping and lifting the object, the robot waited a few seconds to see if the object slips due to bad grasps. A particular grasp was considered a success if the robot was able to lift the object off the table. Our algorithm achieved a success rate of 87%, i.e. 83 out of 95 grasp hypotheses led to a stable grasp. A subset of the achieved grasps[3] is shown in Fig. 7. The results indicate that our template representation is able to generalize from only 15 demonstrations to a large variety of objects. It also indicates that our algorithm is robust against different viewpoints on the object. A few attempts failed due to slightly miss-oriented grasp poses which led to pre-mature contact with the object and made them fall over or roll away. More sensitive closure of the hand can be achieved by exploiting additional features from tactile sensing.

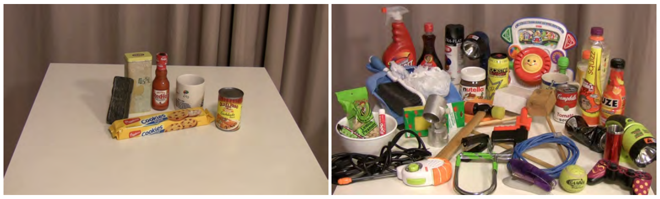

Fig. 5. Seven objects used in the training set (left) and 38 objects used as test set to evaluate the proposed algorithm on the PR2 (right).

We chose a different test set to demonstrate 6 gripper configurations on 4 objects to the Barrett_Hand as seen in Fig. 8. The algorithm was applied to 10 different objects placed in 5 different poses each. Different from the previous experiment, we notified the robot of failed trials. If a grasp failed we positioned it in the same pose for the next trial until it succeeded. The algorithm computed grasp hypotheses that resulted in 41 out of 50 successful trials as shown in Table I. In case of an unsuccessful grasp attempt, the robot achieved to grasp the object in the same pose after at most 2 additional trials using feedback from the previously failed attempts.

Object	Success rate
Spray Can	4/5
Pipe	4/5
Flashlight	4/5
Shovel	3/5
Phone Handle	5/5
Toy Wheel	3/5
Box	4/5
Red Spray Bottle	5/5
Canteen	5/5
Duster with Pan	4/5
Overall	41/50

TABLE I

Grasp attempts on a Barrett WAM arm with Barrett_Hand.

2) Improvement over time: In a further experiment we tried to grasp one object over and over again notifying the algorithm of failed grasps to test if our grasp planner is able to improve over time autonomously. The object was placed in different poses on the table as in the previous experiments. On the PR2 we used a whiteboard marker (see image in the center of the upper row in Fig. 7). The robot tried to grasp it 34 times in a row using the demonstrated grasps from the previous experiment. Although grasping failed quite often in the beginning, we could increase the overall success rate about 30% as shown in Fig. 6. The first trials failed mainly because of grasp poses that led to premature contact with the object. However, for the last trials the PR2 tended to approach the object perpendicular to the table such that premature contact was avoided and the grasps succeeded more often. We could show that our algorithm is able to improve ranking over time using a simple form of feedback. In our experiments a human notified the robot of failed grasp attempts. However, one could think of an autonomous implementation.

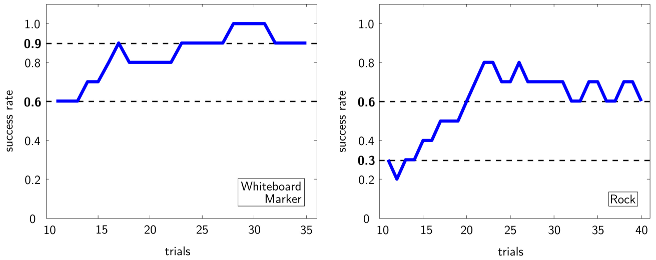

Fig. 6. Success rate with increasing number of grasp trials using feedback from failed attempts. Success rate is computed for the last ten trials at each step. Results on a PR2 and a whiteboard marker (left) and a WAM arm and a rock (right). One can recognize a convergence to suboptimal success rates due to slightly miss-oriented grasp poses and the slippery surface of the rock.

We conducted a similar experiment on the Barrett WAM arm, where the robot tried to grasp a rock shaped object 40 times in a row (see third image from left in the bottom row of Fig. 8). As seen in Fig. 6 we could increase success

[3] Some of the grasp executions can be seen in the video supplement and on http://www.youtube.com/watch?v=noRv2gsy2u0

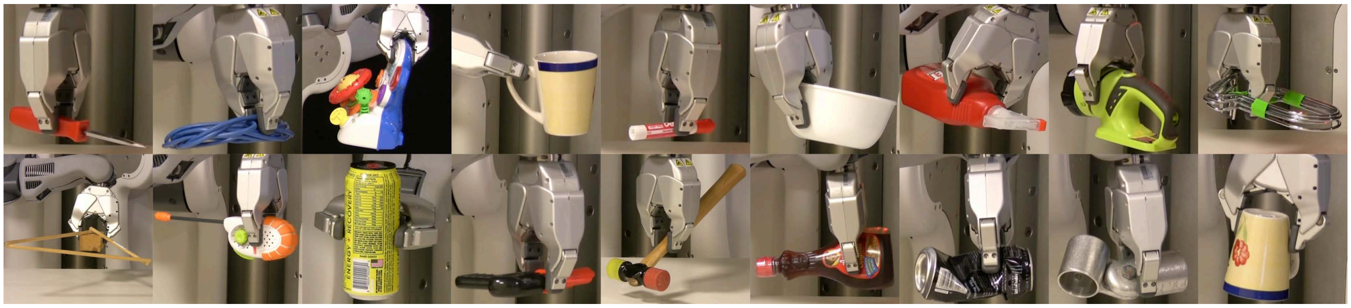

Fig. 7. Subset of the achieved grasps on the test set on the PR2.

rate from 30% to 60%. For the first trials the robot chose the grasp configuration that we taught on the bowl as seen in the bottom left image in Fig. 8. This made the rock slip away quite often. However, the ranking of grasp hypotheses changed over time such that the configuration demonstrated on the leather bag was chosen more often. This lead to more stable grasps and increased the success rate. However, some grasps still failed due to the slippery surface of the object.

Our results show that the presented grasp planner computes grasp hypotheses that lead to successful grasps. We did not need new parameterization of the algorithm to achieve a high success rate on different data sets. The presented grasp planner was tested on two very different real robots with real sensor data. Further, we could show that our algorithm works on two different hand architectures.

As described in Section II-B the computational time of our algorithm depends on the parameters r and n. We set an $r = 16$, and $n = 30$ to keep computational time low. Indeed, we noticed that increasing both values did not increase performance notably. During experiments our algorithm finished computations after 5 to 30 seconds when it ran together with other tasks on the built-in computer on the PR2. Please note that the implementation is not optimized for speed.

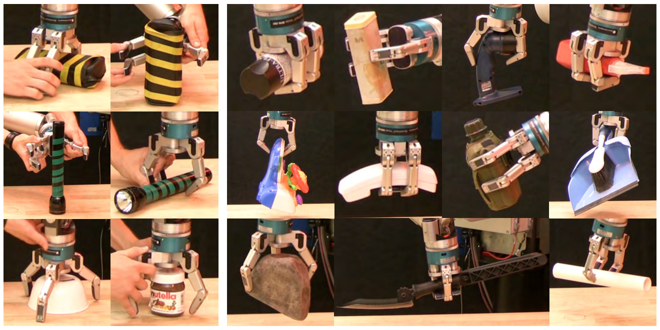

Fig. 8. All demonstrated grasps on a Barrett WAM arm (left) and a subset of the achieved grasps (right).

IV. CONCLUSION

In this paper we presented a template-based grasp selection algorithm which uses demonstrated grasp configurations and generalizes them to grasps for novel objects. We showed on two robots and two different test sets that the proposed template representation is able to capture object features enabling our algorithm to achieve a good success rate on a challenging set of differently shaped objects. New grasps can be taught through kinesthetic teaching to extend the robot's grasp repertoire. Additionally, the proposed method was able to improve the grasp selection process autonomously through trial-and-error. The algorithm has been implemented efficiently and results have been presented on the PR2 and WAM arm.

In future work we will analyze the system in more extend to provide deeper insight. Although we achieved a high success rate in our experiments, our goal is to adapt grasps using tactile sensing in order to prevent slippage and further increase the stability of grasps. Also information about the task to be achieved with an object is to be used in order to provide a more refined grasp selection.

REFERENCES

[1] C. Goldfeder, "Data-driven grasping," Ph.D. dissertation, Columbia University, 2010.

[2] N. Curtis, J. Xiao, and S. Member, "Efficient and effective grasping of novel objects through learning and adapting a knowledge base," in *ICRA*, 2008.

[3] A. Miller and P. Allen, "Graspit! a versatile simulator for robotic grasping," *Robotics & Automation Magazine, IEEE*, vol. 11, pp. 110 – 122, 2004.

[4] M. Przybylski, T. Asfour, and R. Dillmann, "Unions of balls for shape approximation in robot grasping," in *IROS*, 2010.

[5] K. Hsiao, S. Chitta, M. Ciocarlie, and G. E. Jones, "Contact-reactive grasping of objects with partial shape information," in *IROS*, 2010.

[6] E. Klingbeil, D. Drao, B. Carpenter, V. Ganapathi, O. Khatib, and A. Y. Ng, "Grasping with application to an autonomous checkout robot," in *ICRA*, 2011.

[7] R. Detry, E. Başeski, M. Popović, Y. Touati, N. Krüger, O. Kroemer, J. Peters, and J. Piater, "Learning continuous grasp affordances by sensorimotor exploration," in *From Motor Learning to Interaction Learning in Robots*. Springer, 2010.

[8] A. N. Erkan, O. Kroemer, R. Detry, Y. Altun, J. Piater, and J. Peters, "Learning probabilistic discriminative models of grasp affordances under limited supervision," in *IROS*, 2010.

[9] L. Montesano and M. Lopes, "Learning grasping affordances from local visual descriptors," in *Int. Conf. Development and Learning ICDL*, 2009.

[10] A. Saxena, L. L. S. Wong, and A. Y. Ng, "Learning grasp strategies with partial shape information," in *AAAI*, 2008.

[11] J. Bohg and D. Kragic, "Grasping familiar objects using shape context," in *ICAR*, 2009.

[12] M. Kalakrishnan, J. Buchli, P. Pastor, and S. Schaal, "Learning locomotion over rough terrain using terrain templates," in *IROS*, 2009.

[13] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "stomp: stochastic trajectory optimization for motion planning," in *ICRA*, 2011.