Generalizing Pouring Actions Between Objects using Warped Parameters

Sascha Brandl, Oliver Kroemer, and Jan Peters

***Abstract*— One of the key challenges for learning manipulation skills is generalizing between different objects. The robot should adapt both its actions and the task constraints to the geometry of the object being manipulated. In this paper, we propose computing geometric parameters of novel objects by warping known objects to match their shape. We refer to the parameters computed in this manner as *warped parameters*, as they are defined as functions of the warped object's point cloud. The warped parameters form the basis of the features for the motor skill learning process, and they are used to generalize between different objects. The proposed method was successfully evaluated on a pouring task both in simulation and on a real robot.**

I. Introduction

In order to perform tasks in everyday environments, robots will need to be capable of manipulating a wide range of different objects. As objects of the same type may have different shapes and sizes, the robot will have to adapt its actions to the geometry of the specific object that it is manipulating. The shape of objects is particularly important when manipulating liquids, e.g., pouring a glass of water, as liquids conform to the shape of their container. The robot must therefore take into consideration a container's geometry when using it in a pouring task.

Although containers come in a wide variety of shapes and sizes, the important differences can usually be defined by a few geometric parameters [1], [2]. For example, the volume of a container indicates how much fluid it can hold, regardless of whether it has a spherical, or cylindrical shape. A robot can generalize pouring actions between different containers by using these geometric parameters. However, the robot will not be provided with the geometric parameters for most of the novel objects that it encounters. While a human may annotate the geometric information for a couple of objects, the robot will usually need to compute these parameters on its own.

In this paper, we investigate using warped parameters to generalize pouring skills between different objects. A warped parameter is defined as a function on the points of a known object's point cloud. For example, a warped parameter may compute the volume of a set of points' convex hull. When the robot encounters a novel object, it warps the point cloud of the known object to the new object's shape. As a result of the warping, the value of the warped parameter changes to match the geometry of the new object. Once the geometric parameters have been computed, the robot can use them to generalize actions and task constraints between different objects.

All authors are members of the IAS group at the TU Darmstadt, Germany. Jan Peters is also a member of the MPI for Intelligent Systems, Germany. {brandl, kroemer, peters}@ias.tu-darmstadt.de

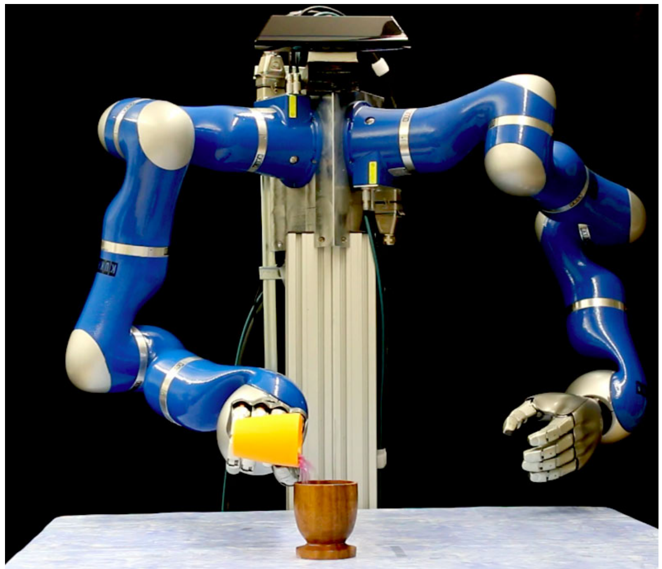

Fig. 1. The robot performs a pouring task with two previously unknown objects. The pouring action was learned from human demonstrations using a taller cup and a wider container to pour into.

In Section II, we explain the process of computing the warped parameters. In Section III, we describe how the robot can learn pouring actions and task constraints that generalize between objects using the warped parameters. The proposed method was successfully evaluated both in simulation and on the robot shown in Fig. 1. The results of the experiments are detailed in Section IV.

Related Work

Several previous works have used warping to generalize manipulations between objects. Hillenbrand et al. [3], [4] used warping to map contact points onto novel objects, in order to transfer grasps between objects. A similar approach was used by Rainer et. al [5], [6] for transferring coordinate frames of task constraints between objects. However, the size and shape of the constraint regions were not adapted to the new object's geometry. Rather than warping only the points on the object, Schulman et al. [7] computed warping functions for the entire scene. The warping was then applied to the demonstrated trajectory of the source scene in order to obtain a trajectory for the current scene. These approaches focus on mapping specific points from the source scene to the target scene, and are therefore especially well-suited for contact-based manipulations. Warped parameters can be used to model more general features of the objects, such as areas and volumes.

Several methods have also been proposed for learning to perform pouring tasks. Pastor et al. [8] learned dynamic motor primitives (DMPs) for pouring from human demonstrations, and used these to generalize to different cup placements. Similarly, Muehlig et al. [9] encoded demonstrated bimanual pouring trajectories using Gaussian mixture models. Rozo et al. [10] proposed learning a controller for pouring tasks based on the observed forces. The work on learning pouring from demonstration has mainly focused on learning with the same set of objects. In comparison, we propose learning in a feature space defined by the warped parameters, in order to automatically generalize between objects.

Some work has also been done on generalizing pouring actions between different objects using reinforcement learning. Kroemer et al. [11] learned a pouring DMP from human demonstrations, and then used a trial-and-error approach to learn the location of a novel container's opening. The opening was detected using a shape-similarity kernel. Tamosiunaite et al. [12] used reinforcement learning to learn the shape of the pouring DMP, as well as the goal point. Reinforcement learning was also used to adapt the learned motion to novel objects, without explicitly considering the differences in geometry.

II. GENERALIZATION WITH WARPED PARAMETERS

In this section, we describe how a robot can compute geometric parameters of an object by warping a known object to match its shape. The object models and the warping process used in this paper are described in Sections II-A to II-C. The computation of the warped parameters for pouring tasks is described in Section II-D.

A. Geometric Object Models

In order to generalize manipulations to a novel object, the robot first computes correspondences between a known source object O_s and the unknown target object O_t. An object O_i is modeled as a set of c_i points located at positions $\boldsymbol{p}_{ij} \in \mathbb{R}^3$ with corresponding normals $\boldsymbol{n}_{ij} \in \mathbb{R}^3$, where $j \in \{1, ..., c_i\}$.

Objects often consist of multiple parts, and a manipulation may only depend on the shape of a part of an object. Hence, geometric parameters often describe the shape of a part rather than the whole object. We therefore also assign each point $\boldsymbol{p}_{ij}$ a vector $\boldsymbol{l}_{ij}$ of length ρ with binary labels, which indicate which of the ρ object parts the point corresponds to. The labels of the target object O_t are initially unknown, but can be computed using the warping process.

An example of an annotated cup can be seen in Fig. 2. The first part is the CONTAINER, which holds the liquids. The second part is the RIM around the opening. We also label the HANDLE as a *dummy* part. As not all containers have handles, it is not used to define any warped parameters for the pouring task, and is only included to help align objects during the warping process.

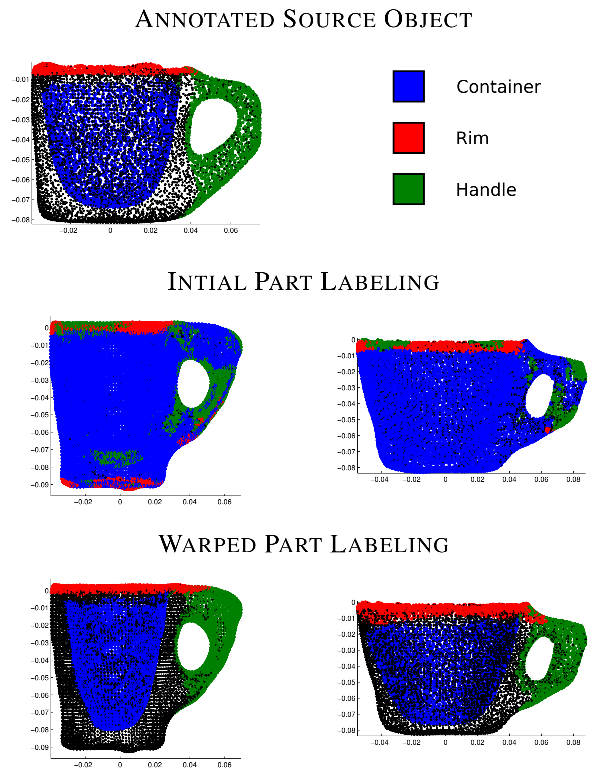

Fig. 2. The *top* row shows the point cloud of the source object, annotated by a human user. The *middle* row shows the point clouds of two target objects. The points were labelled using a classifier based on local features. This intial estimate is only used to compute a coarse alignment with the source object. The point clouds were pre-aligned for this figure to show more clearly how the labels change during the warping process. The *bottom* row shows the final results of the label mapping approach.

B. Warping

Given a source object and a target object, the robot can compute correspondences between the two objects. These correspondences are determined by warping the shape of the source object onto that of the target object. There are various methods for computing 3D warpings between object [13], [14], and the proposed approach does not depend on a specific warping algorithm. We therefore employ a basic warping algorithm for finding correspondences between the containers. The warping process consists of two stages: 1) object alignment, and 2) point mapping

In the first stage, the source object is coarsely aligned with the target object, such that their corresponding object parts are close together. This alignment is accomplished by computing a coordinate system based on the objects' parts. The origin of the coordinate frame is the mean of the container points. The first axis is given by the direction to the mean of the rim points, and the second axis is the orthogonal direction to the mean of the handle points. The third axis is computed by the cross product of the first two axes. As the part labels of the target object $\boldsymbol{l}_t$ are unknown, an initial estimate of the labels is computed using logistic regression. One classifier is trained for each of the three object parts. Each point $\boldsymbol{p}_{ti}$ is classified based on the local distribution of points in its neighborhood. The features used to describe the local distribution of points include the eigenvalues of the covariance matrix, and the distance from the point $\boldsymbol{p}_{ti}$ to the mean of the neighborhood points. The classifiers were trained on the labelled points of the source object. An example of the initial labeling can be seen in Fig. 2. The coordinate frame of the object is estimated using this initial labeling of points. Once the two objects are aligned, the source object was scaled in each direction such that the variances of its container part matched those of the target object. We denote the aligned source objects and target objects by $\tilde{O}_s$ and $\tilde{O}_t$ respectively.

In the second stage of the warping algorithm, the points from the source object $\tilde{O}_s$ are mapped onto the target object $\tilde{O}_t$. This step is similar to the approach proposed by Hillenbrand [15]. Each point of the aligned source object is mapped to the mean of the k nearest neighbors in the aligned target object. In our experiments, we set $k = 1$. Hence, the warped source point $\boldsymbol{p}_{wi}$, with corresponding normal $\boldsymbol{n}_{wi}$ and labels $\boldsymbol{l}_{wi}$, is given by

$$\boldsymbol{p}_{wi} = \boldsymbol{p}_{tj} \text{ , } \boldsymbol{n}_{wi} = \boldsymbol{n}_{tj} \text{ , and } \boldsymbol{l}_{wi} = \boldsymbol{l}_{si},$$
$$\text{s.t. } j = \arg\min \left\|\tilde{\boldsymbol{p}}_{si} - \tilde{\boldsymbol{p}}_{tj}\right\| \text{ and } \tilde{\boldsymbol{n}}_{si}^T \tilde{\boldsymbol{n}}_{tj} > 0.$$

Thus, each source point is mapped to the closest target point with a normal pointing in the same direction. The warped object and its point cloud are denoted by O_w.

C. Point Mapping vs. Label Mapping

The warping process defines a new position and normal for each of the c_s point of the source object O_s. The location of these new points can be used to define warped parameters, as detailed in the next section. We refer to this approach as *point mapping*, as the points of the source object are mapped onto the target object.

However, if the source object has considerably fewer points than the target object, then some details of the target object may not be captured by the warped object. This issue can be addressed by warping the target object to match the source object. The alignment and scaling of the objects is performed as before. However, the points of the target object are mapped onto the source object. The label of each of the target points is then determined using a k-nearest neighbors classifier. In our experiments, we again used $k = 1$, such that

$$\boldsymbol{p}_{wi} = \boldsymbol{p}_{ti} \text{ , } \boldsymbol{n}_{wi} = \boldsymbol{n}_{ti} \text{ , and } \boldsymbol{l}_{wi} = \boldsymbol{l}_{sj},$$
$$\text{s.t. } j = \arg\min \left\|\tilde{\boldsymbol{p}}_{si} - \tilde{\boldsymbol{p}}_{tj}\right\| \text{ and } \tilde{\boldsymbol{n}}_{si}^T \tilde{\boldsymbol{n}}_{tj} > 0.$$

We refer to this approach as label mapping, as the labels of the source object are mapped onto the target object. When using multiple neighbors $k > 1$, the point is assigned to a part if the majority of its k neighbors belong to that part.

The benefit of using the label mapping approach is that it guarantees that all of the points of the target object are used for computing the warped parameters. However, when using label mapping, points can only be referred to by their label and not as individual points. In comparison, when using point mapping, one can refer to individual points, e.g., $\boldsymbol{p}_{w72}$, which correspond to specific points on the source object. The bottom row of Fig. 2 shows an example of using label mapping.

D. Warped Parameters

Having computed the correspondences between the known source object and the novel target object, the robot can compute the warped parameters for the target object. A warped parameter is defined as a function on the warped point cloud $f(O_w)$. Warped parameters can be used to define geometric reference parameters, such as lengths, areas, and volumes, of an object's part. Warped parameters can also be used to define task frames.

For pouring, the task frame is defined by the lip point of the first container, and the center of the second container's opening. The center of the opening is defined as the mean of the rim points. The lip point is defined as the rim point that is the closest to the other container. A pouring motion is defined by the trajectory of the held container's lip point relative to the center of the second container's opening. The trajectory includes the relative 3D position and the tilt of the first container about its lip point. The other two rotation dimensions are usually assumed to be zero. If there is no second container, the lowest rim point is defined as the lip point.

The geometric reference parameters for pouring include the radius of the opening, the volume of the container, the height of the container, and a reference angle for tilting the cup. The radius of the opening is given by the mean distance between the rim points and the center of the opening. The volume of the container is given by the volume of the container points' convex hull. The height of the container is given by the range of all of the points along the first dimension. A tilt reference angle is defined by the amount that the cup must be rotated about the lip point, such that half of the container's volume is above the lip point. As the warping process reshapes the points of the source object, the estimates of the reference parameters will change accordingly. In this manner, the warped parameter function defines how the parameter's value is grounded in the object's geometry.

As the above examples show, warped parameters can be used to define various object properties, and can even build on each other. These parameters can then be automatically computed for new objects using the warping process.

III. Learning with Warped Parameters

In this section, we describe how a robot can learn pouring actions and task constraints that generalize to new objects using the warped parameters.

A. Learning Task Constraints

When performing a pouring task, the liquid should remain in the cup while it is being transported, and it should only be poured out if it will be transferred to another container. These task constraints correspond to phase transitions [16] and can

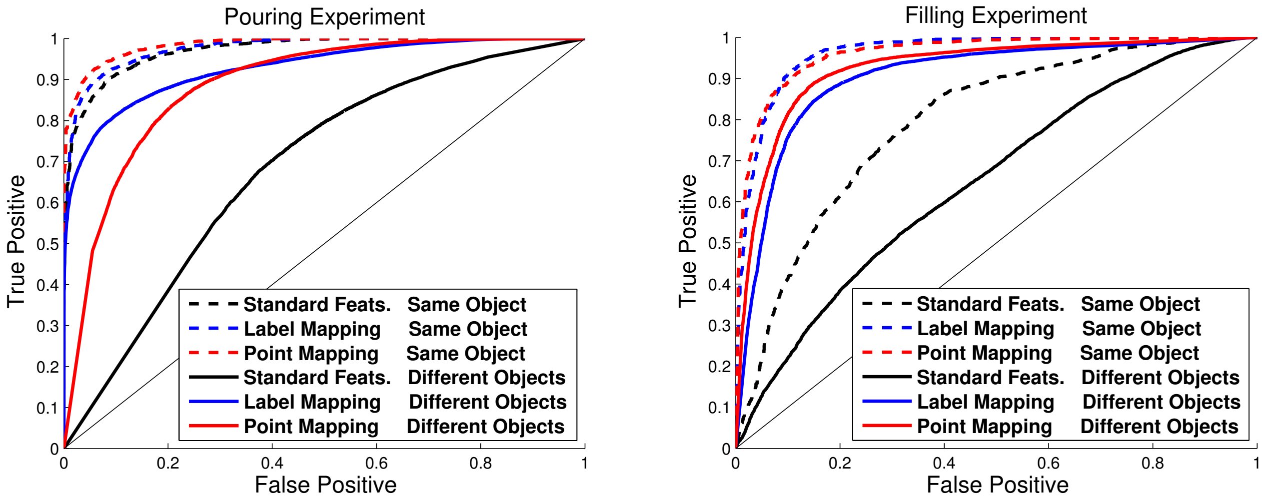

Fig. 3. The figure shows the ROC curves for the learned classifiers for both the pouring experiment and the filling experiment. The *dashed* lines indicate the performance when the classifier is applied to data from the same object that was used for training the classifier. The *solid* lines indicate the performance when the classifiers are applied to novel objects, for which they had no training data. A classifier is generally considered to perform better if it gets closer to the top left corner. Classifiers were trained using features based on the warped parameters computed using both the label mappings and point mappings approaches. The standard features approach did not use the reference values given by the warped parameters.

be fulfilled by learning to predict when the held container will start to pour and when the poured liquid will fill the second container. The conditions for pouring and filling are learned by training a classifier for each condition. The classification is performed using logistic regression, which is a form of probabilistic classifier. The probability of pouring $y_p = 1$ from the first container is given by

$$p(y_p = 1|\boldsymbol{x}_u) = (1 + \exp(-\boldsymbol{\omega}^T \boldsymbol{\varphi}(\boldsymbol{x})))^{-1}$$

where $\boldsymbol{\varphi}(\boldsymbol{x})$ is a vector of features describing the state of the container $\boldsymbol{x}$, and the weight vector $\boldsymbol{\omega}$ is computed from training data using iterative reweighted least squares. The features $\boldsymbol{\varphi}(\boldsymbol{x})$ are of the form α/α_r, where α is a variable and α_r is a reference value defined by a warped parameter. For predicting pouring, the features include the tilt angle of the cup divided by the tilt reference angle, and the fluid volume divided by the volume of the container. The resulting features are dimensionless quantities that automatically adapt to the geometry of the container.

For predicting when the poured liquid increases the fluid volume in the second container $y_f = 1$, we expand the set of features to include both objects and their relative positions. The vertical distance between the containers is divided by the height of the first container. The horizontal distances between the containers are divided by the radius of the second container. These features allow the robot to learn when the poured liquid will miss the second container, as well as predict when the container will overflow.

B. Learning Motor Primitives in Warped Spaces

The proposed warping approach can also be used to learn motor primitives that adapt to the shape of the objects being manipulated. Motor primitives are often used to define desired trajectories that can be easily adapted to different situations. In order to model distributions of trajectories, we use the probabilistic motor primitives (ProMPs) [17]. These motor primitives encode correlations between the different dimensions of the trajectory, and can be conditioned on the initial state of the objects.

The learned motor primitive defines a desired trajectory in the task space described in Section II-D. Similar to the features used to generalize task constraints, the trajectories are defined as dimensionless quantities. The vertical distance between the objects is divided by the height of the held container, and the tilt angle is divided by the reference tilt angle. The horizontal distances are divided by the radius of the second container.

The motor primitives are learned by scaling the demonstrated trajectories according to the warped parameters of the objects used in the demonstrations. In order to execute a pouring action, the robot samples a trajectory from the ProMP, and rescales it according to the current objects' warped parameters.

IV. Experiments

The proposed method was implemented and evaluated both in simulation and on a real robot. The robot, shown in Fig. 1, consists of two Kuka light weight robot arms, each equipped with a five-fingered DLR hand [18]. The robot observes the table-top scene from above using a Microsoft Kinect camera. Ten different cups and bowls were scanned from multiple views. 3D mesh models were generated using an implicit surface representation and marching cubes [19].

A. Simulated Pouring and Filling Experiments

In the first experiment, we evaluated how well task constraints generalize between objects when using warped parameters. The objects were simulated using the Bullet physics engine [20] together with Fluids 2 for incorporating smoothed particle hydrodynamics [21].

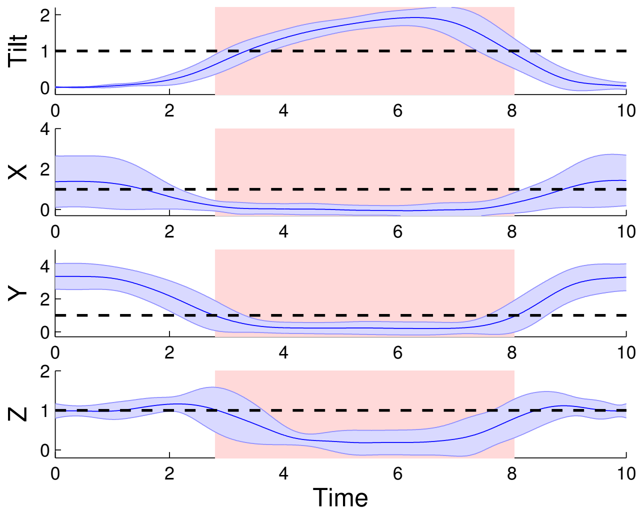

Fig. 4. The plots show the distribution over trajectories learned by the ProMPs in the generalized space. The blue line indicates the mean trajectory, and the shaded regions correspond to +/- two standard deviations. The black horizontal lines indicate when the value is one. The tilt is one when the cup is tilted such that half of the container's volume is above the lip points. The X and Y values are one when the lip point is one radius away from the second container's center. The Z value is one when the vertical distance between the cup and the container is the same as the height of the cup. The red region indicates when the X-Y position of the cup's lip point is within one radius of the container's center.

Each object was filled 1000 times with a random amount of liquid, and tilted by a random angle around the lip point. If the volume of the fluid in the cup decreased, the trial was labelled as pouring $y_p = 1$. Otherwise it was labelled as not pouring $y_p = 0$. The classifiers were trained on sets of 50 samples. The classifiers were tested on two test sets: the 950 other samples from the same object, and the 9000 samples from the other objects. The latter dataset is used to test how well the classifiers generalize between different objects.

A similar procedure was used for the filling experiment. However, the cup used for pouring always contained 10 particles at the start of the trial, and the second container was filled by a random amount. The cup was always tilted by $120°$. The relative positions of the cups were varied between trials. A trial was considered as successful $y_f = 1$ iff none of the particles ended up outside of the second container.

For each training set, three classifiers were computed. The first two classifiers were trained using the warped parameters from the point mapping and the label mapping approaches respectively. The features used for training the classifiers were described in Section III-A. As a benchmark, we also evaluated the classifiers without using the warped parameters. In this case, all of the reference values α_r were set to one, regardless of the objects being manipulated, and the relative positions of the objects were defined by their centers.

The results of the pouring and filling experiments can be seen in Fig. 3. As one would expect, the classifiers generally achieved similar levels of performance when evaluated on the training object. The standard features performed considerably worse in the filling experiment, as different cups were used for pouring even though the second container remained the same. The ROC curves show that the performance of all three classifiers decreases when generalizing to novel objects. However, the drop in performance is considerably less when using the warped parameters. The features based on the warped parameters are therefore better at separating the positive and negative examples across different objects. While the two warping methods performed similarly well on the filling experiment, the label mapping approach performed better in the pouring experiment, detecting more than 50% of the true positives with almost no false positives. The results show that the warping parameters can be used to reliably generalize the constraints of the pouring task between different containers.

B. Robot Pouring Experiment

In the second experiment, the robot used warped parameters to generalize pouring actions between different objects. The robot was provided with ten demonstrations of a pouring task using kinaesthetic teaching. All of the demonstrations were performed with the same two objects shown in the left picture of Fig. 5. For safety reasons, the task was performed with gel balls rather than an actual liquid. The cup was half full at the start of each trial. Using the ten demonstrations, the robot learned a ProMP for pouring, as described in Section III-B. The learned distribution over trajectories is shown in Fig. 4. The robot was then given the task of pouring with different objects. The robot successfully learned to pour from a shorter cup into a bigger bowl, a smaller cup, and a square bowl, as shown in Fig. 5. Only a couple of gel balls were spilled during the experiments. A video of the robot performing the pouring task is also provided in the supplementary material.

As the cups were half-full, pouring usually commenced when the tilt value went above one. Fig. 4 shows that the distribution over trajectories remains safely below this value until the lip point is above the opening. When moving the cup back, most of the liquid has been poured out, and hence the cup can be tilted more. The pictures in Fig. 5 show that the cup was often placed close to the rim of the second container, which indicates that the robot was able to adapt the learned trajectory to the geometry of the object being manipulated.

C. Future Work

The autonomy of the robot can be increased by learning warped parameters. The points could be labelled by using an unsupervised approach to segmenting the objects into parts. A set of generic geometric functions could then be applied to each part in order to generate a library of warped parameters. Feature selection methods could then be applied to select a suitable set of warped parameters for the given task.

The focus in this paper was on learning pouring skills from a single object. The generalization between objects therefore relies on using the warped parameters to construct dimensionless features for the robot. However, given data

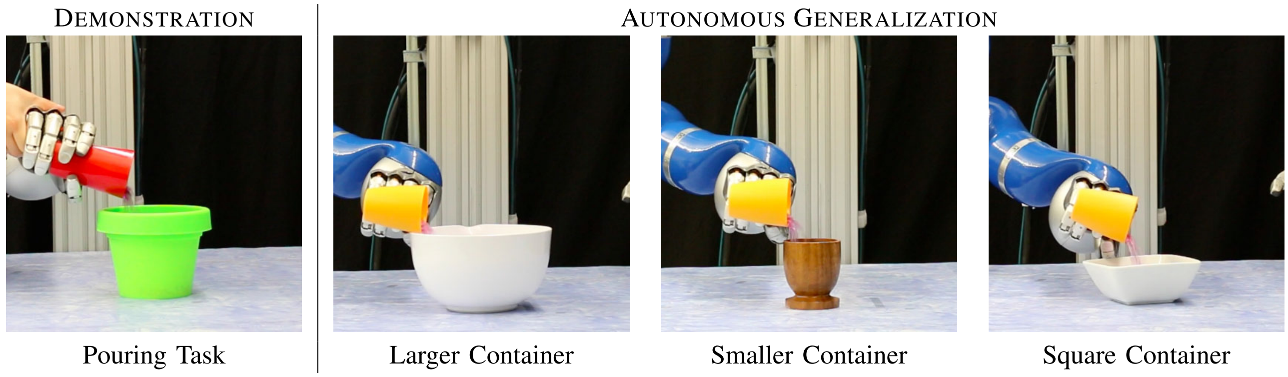

Fig. 5. The pictures show the key results of the real robot experiment. The robot was provided with multiple demonstrations of the pouring task using kinaesthetic teaching, as shown on the left. Using the warped parameters approach, the robot successfully generalized the demonstrated actions to novel objects with different shapes and sizes, as shown on the right.

from multiple objects, the robot can also learn how to generalize between objects. In this case, the warped parameters could be treated as separate features that describe the object. For example, ProMPs can be used to learn the correlations between the trajectory parameters and the warped object parameters. Object-specific trajectories can be obtained by conditioning on the current object parameters. This approach would even allow the robot to learn that only some segments of the trajectory depend on the object parameters. However, learning ProMPs in this manner would require additional training trajectories with different objects. These trajectories could be obtained from human demonstrations, or by adapting trajectories using reinforcement learning [12].

V. CONCLUSION

We proposed using warped parameters to generalize pouring skills between different objects. Warped parameters are functions defined on the point cloud of a known object. The parameter can be computed for a novel object by warping the known object's point cloud to match the geometry of the novel object.

The proposed method was successfully evaluated both in simulation and on a real robot pouring task. The experiments showed that the warped parameters can be used to generalize task constraints and motor primitives between containers of different shapes and sizes

ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7-ICT-2013-10) under grant agreements 248273 (GeRT), 270327 (CompLACS), and 610878 (3rdHand). The authors would also like to thank Ullrich Hillenbrand for the insightful discussions.

REFERENCES

[1] G. Bartels, I. Kresse, and M. Beetz, "Constraint-based movement representation grounded in geometric features," in *International Conference on Humanoid Robots*, (Atlanta, Georgia, USA), 2013.

[2] M. Tenorth, S. Profanter, F. Balint-Benczedi, and M. Beetz, "Decomposing cad models of objects of daily use and reasoning about their functional parts," in *International Conference on Intelligent Robots and Systems*, pp. 5943–5949, 2013.

[3] U. Hillenbrand and M. Roa, "Transferring functional grasps through contact warping and local replanning," in *International Conference on Intelligent Robots and Systems*, pp. 2963–2970, Oct 2012.

[4] H. Ben Amor, O. Kroemer, U. Hillenbrand, G. Neumann, and J. Peters, "Generalization of human grasping for multi-fingered robot hands," in *International Conference on Intelligent Robots and Systems*, 2012.

[5] R. Jäkel, *Learning of Generalized Manipulation Strategies in Service Robotics*. PhD thesis, Institut für Anthropomatik, Karlsruhe, 2013.

[6] R. Jäkel, S. Schmidt-Rohr, S. Rühl, A. Kasper, Z. Xue, and R. Dillmann, "Learning of planning models for dexterous manipulation based on human demonstrations," *International Journal of Social Robotics*, vol. 4, no. 4, pp. 437–448, 2012.

[7] J. Schulman, J. Ho, C. Lee, and P. Abbeel, "Learning from demonstrations through the use of non-rigid registration," in *International Symposium on Robotics Research*, 2013.

[8] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal, "Learning and generalization of motor skills by learning from demonstration," in *International Conference on Robotics and Automation*, 2009.

[9] M. Muehlig, M. Gienger, S. Hellbach, J. J. Steil, and C. Goerick, "Task-level imitation learning using variance-based movement optimization," in *International Conference on Robotics and Automation*, (Kobe, Japan), pp. 1177–1184, IEEE, 05/2009 2009.

[10] L. Rozo, P. Jimenez, and C. Torras, "Force-based robot learning of pouring skills using parametric hidden markov models," in *Robot Motion and Control*, pp. 227–232, July 2013.

[11] O. Kroemer, E. Ugur, E. Oztop, and J. Peters, "A kernel-based approach to direct action perception," in *International Conference on Robotics and Automation*, 2012.

[12] M. Tamosiunaite, B. Nemec, A. Ude, and F. Wörgötter, "Learning to pour with a robot arm combining goal and shape learning for dynamic movement primitives," *Robotics and Autonomous Systems*, 2011.

[13] A. W. F. Lee, D. Dobkin, W. Sweldens, and P. Schröder, "Multiresolution mesh morphing," in *Annual Conference on Computer Graphics and Interactive Techniques*, (New York, NY, USA), pp. 343–350, 1999.

[14] F. Steinke, B. Schölkopf, and V. Blanz, "Learning dense 3d correspondence.," in *Advances in Neural Information Processing Systems*, pp. 1313–1320, MIT Press, 2006.

[15] U. Hillenbrand, "Non-parametric 3d shape warping," in *International Conference on Pattern Recognition*, pp. 2656–2659, IEEE, 2010.

[16] O. Kroemer, H. van Hoof, G. Neumann, and J. Peters, "Learning to predict phases of manipulation tasks as hidden states," in *International Conference on Robotics and Automation*, 2014.

[17] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, "Probabilistic movement primitives," in *Advances in Neural Information Processing Systems*, 2013.

[18] Z. Chen, N. Y. Lii, T. Wimboeck, S. Fan, M. Jin, C. Borst, and H. Liu, "Experimental study on impedance control for the five-finger dexterous robot hand dlr-hit ii.," in *International Conference on Intelligent Robots and Systems*, pp. 5867–5874, IEEE, 2010.

[19] S. Dragiev, M. Toussaint, and M. Gienger, "Gaussian process implicit surfaces for shape estimation and grasping," in *International Conference on Robotics and Automation*, 2011.

[20] "Bullet physics library," October 2013.

[21] "Bullet-fluids project," October 2013.