Learning Manipulation Actions from Human Demonstrations

Tim Welschehold Christian Dornhege Wolfram Burgard

***Abstract*— Learning from demonstration is a popular approach for teaching robots as it allows service robots to acquire new skills without explicit programming. However, for manipulation actions mostly kinesthetic teaching is used as these actions require precise knowledge about the interactions between the robot and the object. In this paper, we present a novel approach that allows a robot to learn actions carried out by a teacher from observations. We achieve this by first transforming RGBD observations to consistent hand-object trajectories, which are then adapted to the robot's grasping capabilities. Experimental results show that the robot is able to learn complex tasks such as opening doors or drawers.**

I. Introduction

Service robots acting in households are expected to interact with a multitude of objects. Obviously, it is not feasible to pre-program all necessary manipulation skills, especially for new objects that a human wants to be handled in a certain way. Learning by demonstration is a promising approach for teaching new skills to a robot. However, doing so with kinesthetic teaching requires that the human can physically guide the robot and has a sufficient understanding of the robot's grasping capabilities. Therefore, we present a system that, in contrast to other approaches, learns manipulation actions from human demonstrations, where a task is performed by the teacher whom the robot only observes with its onboard sensors. The envisioned system is illustrated in Fig. 1.

One of the major challenges in this context lies in the fact that the demonstrations provided by the human teacher typically cannot be executed exactly by the robot, due to different kinematics and grasping capabilities. Grasp planning, although able to provide high-quality grasps, is not a sufficient solution as human demonstrations also express preferences. For example a glass should not be grasped from the edge as the gripper would touch the inside. To achieve a successful reproduction of the intended object manipulation one must therefore balance between precisely following the hand trajectory, i.e., imitating the human demonstration and adapting this trajectory to the robot's grasping capabilities. A further problem comes from the fact that the robot only watches the scene from a single perspective—its own. Occlusions or bad viewing angles might temporarily make the robot loose track of the human or the object.

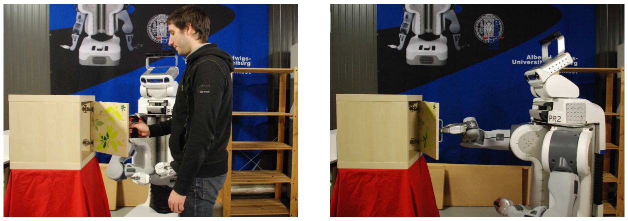

Fig. 1: Learning from human demonstrations: The robot observes the human demonstrating a task, adapts this to its own capabilities and executes the task on its own.

In this paper we present a novel approach to teaching robots manipulation actions by solely observing a human. In particular, we make the following contributions. We fill potential gaps in hand-object trajectories for a manipulation task. Based on the assumption that during manipulation the grasp stays fixed, i.e., the transformation from the object to the hand does not change, we pose a graph optimization problem that leads to consistent hand-object trajectories. We adapt human hand motions to feasible gripper motions based on the fact that, although the robot does not know how to exactly handle the object in question, it does know its own grasping capabilities. To this end, we formulate an objective function that, for a given pose, considers on the one hand the distance to the demonstrations and on the other hand the quality of the pose as a grasp for the object. By minimizing this function, we obtain the best grasp pose starting from which we calculate appropriate gripper-object trajectories.

Our approach has been implemented and tested on a PR2 robot in a real-world environment for challenging tasks including moving objects and opening doors or drawers. The experiments demonstrate that our robot is able to precisely execute even constrained tasks learned from a low number of human demonstrations observed with an RGBD camera.

II. Related Work

In the field of learning from demonstration approaches using dynamical systems for motion representation like [1]–[3] have shown promising results over the last years. The main focus of these contributions is on motion generation and execution. Pastor *et al.* [2] learn a non-linear differential equation to reproduce demonstrated movement. Their work is based on the dynamic movement primitive (DMP) framework [4]. Calinon *et al.* [1] extend the concept of the DMP by formulating the motion mechanism as driven by a virtual spring-damper system. Their work proposes a probabilistic approach that estimates the dynamical system's parameters in a Gaussian mixture regression.

While those approaches achieve good results in motion generation, they do not deal with human demonstrations of object manipulation tasks. Often these are performed by kinesthetic teaching [3], [5]. Pastor *et al.* [2] train motion with an exoskeleton arm and define goal points on the

All authors are with the Institute of Computer Science, University of Freiburg, Germany. This work has been supported by the Baden-Württemberg Stiftung (project ROTAH) and the German Research Foundation under research unit BU 865/8-1 (HYBRIS).

manipulated objects manually. Calinon *et al.* [1] track human hand trajectories with markers, but do not address manipulation tasks. Mühlig *et al.* [6] and Asfour *et al.* [7] present systems that learn from human teachers using articulated models of a human body. Although Mühlig *et al.* deal with manipulation tasks, grasping is not part of the system and is either preprogrammed or assumed as given. Asfour *et al.* do not perform a motion transfer to a robotic platform. Work by Lioutikov *et al.* [8] also handles imitation learning but focuses on segmentation of the demonstrated tasks while grasp adaption is not addressed.

Another approach to robust object handling is to use grasp planning. The focus here lies on the detection of stable grasps of objects. Ten Pas *et al.* [9] propose a method to detect grasps on unknown objects in cluttered scenes using RGBD data. They sample a set of grasp hypotheses and classify them with machine learning techniques. Work by Saxena *et al.* [10] aims at finding grasps relying only on 2D images of objects. A simulated labeled dataset is generated and used to train a probabilistic model, which identifies suitable grasping regions based on visual features. GraspIt! [11] is used to find stable grasps based on robot gripper and object models.

In contrast to these robot based procedures there are also human inspired grasp approaches. Gioioso *et al.* [12] map the human hand to robot grippers exploiting synergetic composition of the human hand. Lin and Sun [13] present a grasp planning approach that builds upon human demonstrated grasp types. The work of Armor *et al.* [14] aims at generalizing human grasps. They propose a direct mapping of human hand joint angles to the robot and achieve generalization by adapting demonstrated contact points to unknown objects. In contrast to our work these methods use precise hand postures for training and focus on grasping itself and not on manipulating objects. Our work complements these approaches as we are not only interested in optimal grasps, but aim at choosing a grasp that balances grasp quality and the resemblance to the demonstrated manipulation task.

III. Problem Statement

A demonstration of a manipulation action consists of two 6-dof trajectories: one for the teacher's hand and one for the manipulated object. A trajectory $\boldsymbol{X}$ is a sequence of n poses $\boldsymbol{X} = \langle \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \rangle$. Hand and Object trajectories are not required to be the same length and can contain gaps due to missing observations. A pose $\boldsymbol{x}$ is represented by a translational part $pos(\boldsymbol{x})$ given by a 3d vector $(x, y, z)^T$ and a rotational part $rot(\boldsymbol{x})$ given by a quaternion $(q_x, q_y, q_z, q_w)^T$. Additionally each pose $\boldsymbol{x}$ has a time stamp $time(\boldsymbol{x})$ to associate matching poses.

The demonstrated trajectories contain reach and retreat parts, where the object is not moving, and manipulation parts, where object and hand move together in a fixed grasp. Note that this assumption is necessary since our robot cannot perform in-hand manipulations. For more elaborate grippers this assumption might be relaxed.

Expert demonstrations, in which the human hand trajectory precisely leads to a suitable grasp for the robot are not required. Furthermore, no offset from the hand to the gripper is given. The goal is to learn a motion model that allows the robot to reproduce the intended action from the human.

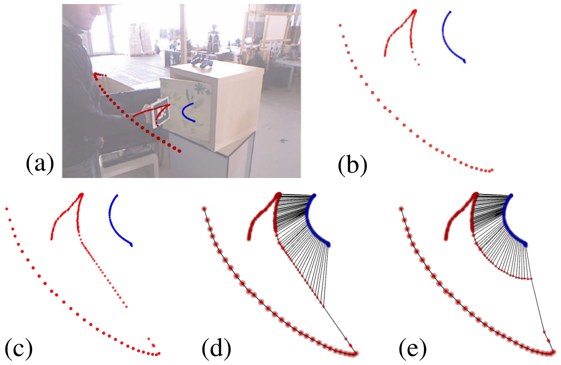

Fig. 2: Acquiring consistent hand (red) and object (blue) trajectories for opening a door (a): The top-down projection of the raw data (b) shows a large gap, where the hand was not tracked. Missing points are filled by interpolation (c). A graph optimization problem is posed with the constraint that during manipulation the hand-object transformation is fixed (d). The correction results in consistent trajectories (e).

IV. Consistent Hand-Object Trajectories

In this section we show how to generate dense, consistent pairs of hand and object trajectory segments that are suited for motion learning. Observations are made by a robot with its RGBD camera. Besides noisy data and errors from hand and object detection algorithms, we face two general problems: First, the relative measured pose of the object in the hand might change during the demonstration. Second, observations might contain gaps, where either the hand or object is obstructed from the robot's view. To make the data suitable for learning a motion model, we first segment raw hand and object trajectories and then in a graph optimization step correct the grasp transformation and fill the observation gaps.

A. Acquiring Hand and Object Trajectories

In order to teach the robot an object related task we need to track both the human hand and the involved objects during demonstrations. Any algorithm that produces tracks of hand or object poses is applicable. Tracking the hand is currently achieved using a marker attached to the hand. For object tracking we use SimTrack [15]. This gives us the two trajectories $\boldsymbol{X}_h^R$ for the hand poses and $\boldsymbol{X}_o^R$ for the object poses with raw tracking data. As a first step, we compute matching same-length trajectories $\boldsymbol{X}_h$ and $\boldsymbol{X}_o$, where for each $\boldsymbol{x}_i^h \in \boldsymbol{X}_h$ there is a respective $\boldsymbol{x}_i^o \in \boldsymbol{X}_o$ with $time(\boldsymbol{x}_i^h) = time(\boldsymbol{x}_i^o)$ and vice-versa. Whenever for a pose with time t in one trajectory there is no matching one with the same time in the other, we create a new pose with time t in the other trajectory by linear interpolation for the translational part and slerp for the rotational part. This leaves us with hand and object trajectories that possibly contain sequences of interpolated poses representing gaps in the observations. An example is visualized in Fig. 2(a)-(c).

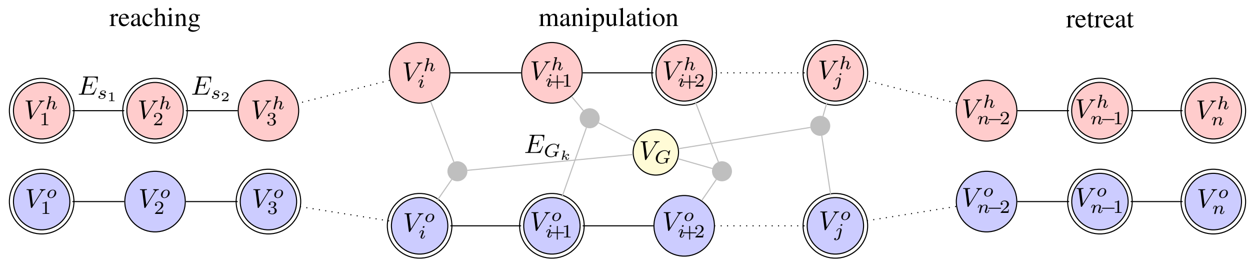

Fig. 3: A hyper-graph is constructed to compute the best estimate from the observations. Vertices for poses from the hand V^h are shown in red, vertices for object poses V^o are shown in blue. Vertices with unary edges that link matched poses to a fixed frame are shown with double circles. Binary edges representing the smoothness of trajectories are shown as lines between vertices and three lines connected by a dot display ternary edges that estimate a single grasp transformation V_G.

B. Segmentation

We segment the hand and object trajectories by labeling matching poses in both trajectories to belong to a reaching, manipulation or retreat segment. We identify manipulation segments by searching for joint hand and object motion. For this we use the co-occurrence of parallel movements [16] defined by the scalar product of hand and object velocities. Whenever this scalar product is greater than the threshold $\delta p{=}0.002m^2/s^4$, hand and object move in the same direction and both velocities are greater than zero, i.e., the object is being manipulated.

C. Trajectory Correction

We now have two labeled trajectories $\boldsymbol{X}_h$ and $\boldsymbol{X}_o$. Both might contain gaps with interpolated poses that do not form consistent manipulation segments, where the hand and object move together. Besides the actual observations contained in the trajectories, we have two other sources of background information: First, trajectories are smooth in the sense that consecutive poses are near to each other. Second, during manipulation the relative pose of hand and object is fixed. We now formulate a hyper-graph optimization problem that simultaneously captures the background information and the observations in a single system. Its solution are two corrected trajectories $\hat{\boldsymbol{X}}_h$ and $\hat{\boldsymbol{X}}_o$ that best represent the hand and object motions given the observations.

We use the general graph optimization library g^{2}o [17] to compute a least-squares solution. Each k-edge in the graph relates its vertices' estimates $\hat{\boldsymbol{x}}_1, \ldots, \hat{\boldsymbol{x}}_k$ with a measurement $z_{1,\ldots,k}$ given for the edge by an error function $\mathbf{e} \overset{\text{def}}{=} \mathbf{e}(\hat{\boldsymbol{x}}_1, \ldots, \hat{\boldsymbol{x}}_k, z_{1,\ldots,k})$. Together with the information matrix $\boldsymbol{\Omega}$ the sum of squared error $\mathbf{e}^T\boldsymbol{\Omega}\mathbf{e}$ for all edges is minimized. This means that besides the structure of the graph, we only need to state, how the error functions are defined and what the information matrices represent. In our case all vertices' estimates are 6-dof poses or transformations and the error functions also compute 6-dof transformations. We represent these internally in the same way as Kümmerle et al. [17].

Fig. 3 shows an illustration of the constructed graph. For each pose $\boldsymbol{x}_i^h \in \boldsymbol{X}_h$ and each pose $\boldsymbol{x}_i^o \in \boldsymbol{X}_o$ a vertex is added representing the estimates $\hat{\boldsymbol{x}}_i^h$ and $\hat{\boldsymbol{x}}_i^o$, respectively. The poses $\boldsymbol{x}_i^h$, $\boldsymbol{x}_i^o$ are also used to initialize their estimates $\hat{\boldsymbol{x}}_i^h$ and $\hat{\boldsymbol{x}}_i^o$. We use unary edges for all vertices from non-interpolated poses. The pose $\boldsymbol{x}_i$ from the trajectories is used as the measurement for the estimate $\hat{\boldsymbol{x}}_i$. The corresponding error function is therefore

$$\mathbf{e}(\hat{\boldsymbol{x}}_i, \boldsymbol{x}_i) = \boldsymbol{x}_i^{-1} \cdot \hat{\boldsymbol{x}}_i \qquad (1)$$

and ensures that the estimates are near the actual observations. The information matrix is set to the accuracy of the object or hand detection algorithm. Between consecutive nodes for $\hat{\boldsymbol{x}}_i$ and $\hat{\boldsymbol{x}}_j$ along each trajectory binary edges E_s are added. These represent the smoothness of the trajectory and keep the relative transformation between poses similar to that of the input trajectories' poses $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ by

$$\mathbf{e}(\hat{\boldsymbol{x}_i}, \hat{\boldsymbol{x}_j}, \boldsymbol{x}_i, \boldsymbol{x}_j) = \left(\boldsymbol{x}_i^{-1} \cdot \boldsymbol{x}_j\right)^{-1} \cdot \left(\hat{\boldsymbol{x}}_i^{-1} \cdot \hat{\boldsymbol{x}}_j\right). \qquad (2)$$

We set the information matrix to reflect the smoothness that we assume for user demonstrations scaled by the distance between the poses. Finally, we address the fact that in the manipulation segment the grasp is a fixed transformation. We introduce an additional node V_G into the graph that estimates the grasp transformation, i.e., the pose of the hand in the object's frame $\hat{\boldsymbol{x}}_h^o$. For each pose pair $\boldsymbol{x}_i^h$ and $\boldsymbol{x}_i^o$ a ternary edge E_G is added that connects the vertices for $\hat{\boldsymbol{x}}_i^h$, $\hat{\boldsymbol{x}}_i^o$ and V_G. The error function

$$\mathbf{e}(\hat{\boldsymbol{x}}_i^h, \hat{\boldsymbol{x}}_i^o, \hat{\boldsymbol{x}}_h^o) \;\;=\;\; \hat{\boldsymbol{x}}_h^{o^{-1}} \cdot \left(\hat{\boldsymbol{x}}_i^{o^{-1}} \cdot \hat{\boldsymbol{x}}_i^h\right) \qquad (3)$$

computes the difference between the grasp estimate $\hat{\boldsymbol{x}}_h^o$ and the relative transformation between the pose estimates for the hand and object. When the error is minimized, the grasp estimate $\hat{\boldsymbol{x}}_h^o$ is as close as possible to the relative transformation. Thus, for all manipulation pose pairs, the error of each relative transformation to the grasp estimate is minimized. By introducing V_G and ternary edges in comparison to just adding binary edges between each pose pair, we do not have to input a grasp transformation into the system. Instead, this is estimated as part of the optimization process. An illustration of the optimization process and its results is shown in Fig. 2(d) and (e). Here, V_G in ternary constraints is not shown as that represents a relative transformation, but not an absolute location.

V. Adapting Human Demonstrations to Robot Capabilities

We now have consistent hand and object trajectories with $\hat{\boldsymbol{X}}_h$ and $\hat{\boldsymbol{X}}_o$. Based on these, we first compute a grasp pose respecting the robot's capabilities and the demonstrations and then apply this to gain gripper trajectories. These are then used in motion learning, so that the resulting model is suitable for execution on the robot.

A. Grasp Adaption

A grasp is given by the pose of the hand or gripper relative to the regarded object. The set of demonstrated grasp poses for a pair of corrected trajectories $\hat{\boldsymbol{X}}_h$, $\hat{\boldsymbol{X}}_o$ is therefore

$$\hat{\boldsymbol{X}}_G = \{\hat{\boldsymbol{x}}_i^{o^{-1}} \cdot \hat{\boldsymbol{x}}_i^h \mid \hat{\boldsymbol{x}}_i^o \in \hat{\boldsymbol{X}}_o, \hat{\boldsymbol{x}}_i^h \in \hat{\boldsymbol{X}}_h\}, \tag{4}$$

where only the indices i that were labeled as manipulation poses are chosen. This set is now used to find the robot grasp. The distance of a pose $\boldsymbol{x}$ to the demonstration $\hat{\boldsymbol{X}}_G$ is defined by

$$d_d(\boldsymbol{x}, \hat{\boldsymbol{X}}_G) = \frac{1}{|\hat{\boldsymbol{X}}_G|} \sum_{\boldsymbol{x}_G \in \hat{\boldsymbol{X}}_G} d_t(\boldsymbol{x}, \boldsymbol{x}_G) + \beta \cdot d_r(\boldsymbol{x}, \boldsymbol{x}_G), \tag{5}$$

with the translational and rotational distances

$$d_t(\boldsymbol{x}, \boldsymbol{x}_G) = \|pos(\boldsymbol{x}) - pos(\boldsymbol{x}_G)\|_2 \tag{6}$$

$$d_r(\boldsymbol{x}, \boldsymbol{x}_G) = 1 - \langle rot(\boldsymbol{x}), rot(\boldsymbol{x}_G)\rangle^2, \tag{7}$$

where $\langle \cdot, \cdot \rangle$ computes the dot product of the quaternions and the parameter $\beta{=}10$ adjusts translational versus rotational error. The best grasp $\boldsymbol{x}_g^*$ is found by minimizing the objective function of the non-linear optimization problem

$$\boldsymbol{x}_g^* = \min_{\boldsymbol{x}} \Psi(\boldsymbol{x}), \text{ such that} \begin{cases} c(\boldsymbol{x}) = 0 \\ \boldsymbol{x}_{min} \leq \boldsymbol{x} \leq \boldsymbol{x}_{max} \end{cases} \tag{8}$$

with

$$\Psi(\boldsymbol{x}) = w \cdot d_g(\boldsymbol{x}, o) + (1 - w) \cdot d_d(\boldsymbol{x}, \hat{\boldsymbol{X}}_G). \tag{9}$$

The objective function $\Psi(\boldsymbol{x})$ consists of two competing parts $d_g(\boldsymbol{x}, o)$ and $d_d(\boldsymbol{x}, \hat{\boldsymbol{X}}_G)$. The weight w allows the user to adjust the procedure to emphasize grasp quality in $d_g(\boldsymbol{x}, o)$ or closeness to the human demonstrations $d_d(\boldsymbol{x}, \hat{\boldsymbol{X}}_G)$. We used $w = 0.5$ in our experiments. The robot's grasping capability is expressed by a grasp quality function $d_g(\boldsymbol{x}_g, o)$ that accepts a potential grasp pose $\boldsymbol{x}_g$ and a representation of the object o, usually a mesh or point cloud. The function states, how far off the pose $\boldsymbol{x}_g$ is from a good grasp for o. The constraints $c(\boldsymbol{x})$ ensure that the quaternion in the rotational part $rot(\boldsymbol{x})$ is normalized and that the pose $\boldsymbol{x}$ is collision free. For efficiency, we first evaluate Ψ on discretized poses in a 2 cm grid and then compute $\boldsymbol{x}_g^*$ with the initial guess from the four best poses using a solver based on interior point techniques [18] in Matlab. The bounds $\boldsymbol{x}_{min}$ and $\boldsymbol{x}_{max}$ for optimization are then set to not deviate more than 3 cm for $pos(\boldsymbol{x})$ and ± 0.5 for $rot(\boldsymbol{x})$. The optimization takes between one and four minutes. An example of the optimization result is shown in Fig. 4.

The grasp quality measure depends on the robot's gripper and object geometry. We use the same mesh as in the object detection in Sec. IV-A as the object representation. In principle any grasp quality function can be used. As our robot is equipped with a simple two finger parallel gripper we use the distance to the gripper's tool center point for grasp quality. With the pose of the tool center point in the gripper frame as $\boldsymbol{x}^t$, we define

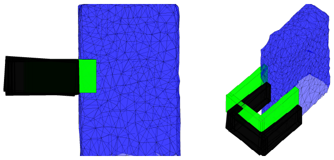

Fig. 4: Grasp adaption for the cereal box (blue) results in a suitable grasp $\boldsymbol{x}_g^*$ (green). Note that in the sideways direction the grasp had to be significantly adapted from the demonstrations (black), while the optimization kept close to the demonstrations in the height.

$$d_g(\boldsymbol{x}, o) = \begin{cases} \|pos(\boldsymbol{x} \cdot \boldsymbol{x}^t) - \boldsymbol{p}_c\|_2, & \text{if } o \text{ in gripper} \\ \|pos(\boldsymbol{x} \cdot \boldsymbol{x}^t) - \boldsymbol{p}_o\|_2 + R, & \text{otherwise} \end{cases} \tag{10}$$

To compute if the object o is inside the gripper, we span a plane by the two finger tips and the tool center point and consider the projection of all points of the object, i.e., its mesh's vertices, onto this plane. If any projection lies inside the convex hull of the gripper on this plane and the respective point is no farther away from the plane than the gripper thickness, we consider the object inside the gripper. $\boldsymbol{p}_c$ is the mean of the projected points inside the plane. Thus, the first condition favors grasps that keep the object in the center of the gripper. $\boldsymbol{p}_o$ is the closest center of a face of the object mesh to the gripper's tool center point. The second condition pushes the object towards the gripper. R is a parameter used to punish the object being outside the gripper. In our experiments we used $R{=}15$. In contrast to choosing large values for grasp poses that do not enclose the object, this guides gradient-based optimization methods towards suitable grasps. This heuristic is inspired by [19]. More complex gripper geometries might require more elaborate heuristics. If no grasp quality measure is available, state-of-the-art grasp planners [9]–[11] can be leveraged by sampling good grasps and computing the distance of $\boldsymbol{x}$ to the closest grasp.

B. From Human to Robot Trajectories

Once a suitable grasp $\boldsymbol{x}_g^*$ is found the hand and object trajectories $\hat{\boldsymbol{X}}_h$ and $\hat{\boldsymbol{X}}_o$ are used to create a robot gripper trajectory. We separate between poses in reaching or retreat segments and poses in manipulation segments. As $\boldsymbol{x}_g^*$ is the best gripper pose in the object's frame, during manipulation the resulting robot poses are derived from applying this grasp to the object poses, i.e.,

$$\boldsymbol{x}_{i_m}^r = \boldsymbol{x}_{i_m}^o \cdot \boldsymbol{x}_g^* \text{ for all } \boldsymbol{x}_{i_m}^o \in \hat{\boldsymbol{X}}_o, \tag{11}$$

if i_m is labeled as a manipulation segment.

For reaching or retreat segments, we do not have a grasp constraint. However, a reaching or retreat motion must connect with a manipulation motion. Thus we estimate the offset between hand and gripper at the transition points between manipulation and reaching or retreat. Then we transform

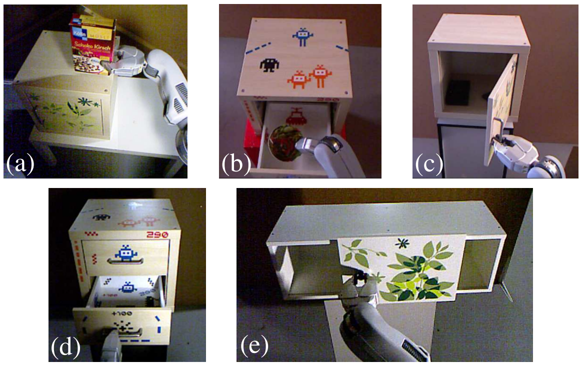

Fig. 5: The robot's view, when repeating the following tasks after demonstration: Placing a cereal box (a), putting a bowl in a drawer (b), opening a swivel door (c), opening a drawer (d) and opening a sliding door (e).

the reaching and retreat motion for the gripper to connect seamlessly to the manipulation segment. The resulting poses together with the $\boldsymbol{x}^r_{i_m}$ form the robot gripper trajectory $\hat{\boldsymbol{X}}_r$.

C. Generating Robot Motions

A corrected gripper trajectory $\hat{\boldsymbol{X}}_r$ is obtained for each demonstration. Multiple demonstrations are merged and independent time-driven models for the reaching, manipulation and retreat parts are learned using mixtures of Gaussians [1]. Alternative approaches for time independent motion learning could also be used [20]. The motion model is designed to generate online motions. As we are only interested in the applicability of the gripper trajectories, we use the learned models to simulate trajectories in the corresponding object frame and execute these on the robot.

VI. Experiments

To evaluate our approach we recorded human demonstrations for five scenarios consisting of moving a cereal box onto a shelf, placing a bowl in a drawer, opening a swivel door, opening a drawer and opening a sliding door. We refer to those as Cereal, Bowl, Door, Drawer and Slide. Examples of the robot executing these tasks are shown in Fig. 5. While the first two are mainly a grasping challenge, the last three tasks in addition require a correct execution of the manipulation trajectory as the object's movements are constrained by geometry.

All demonstrations were recorded with our PR2 robot's RGBD camera with approximately $30Hz$ in a setup as shown in Fig. 1. The data was then processed offline to generate the motion models. These were then executed in real-world experiments on the robot to determine their success. We considered an execution a failure, if the robot failed to grasp the object, collided unintentionally (e.g., bumped into a door, when trying to grasp the handle) or failed to execute the manipulation.

A. Learning from Human Demonstrations

First, we determine, if our approach allows the robot to successfully reproduce human demonstrations. To this end we performed ten executions for each of the five demonstrated tasks: five with the model learned from our approach, i.e., with trajectory correction and grasp adaption, and five, where the model was learned from uncorrected tracking data without grasp adaption, i.e., $\boldsymbol{X}_h$. None of the latter runs lead to a successful execution, while all executions with our approach worked. Examples of the robot performing the tasks are displayed in Fig. 5. While this shows that our approach works in principle, Fig. 6 illustrates, why it is necessary. In most cases, the robot failed to grasp the object, because the human hand pose does not accurately reflect a proper grasp for the robot. Although learning from hand observations has been shown to work for pointing motions and gestures, interacting with objects directly requires accurate models adapted to the robot, which we provide.

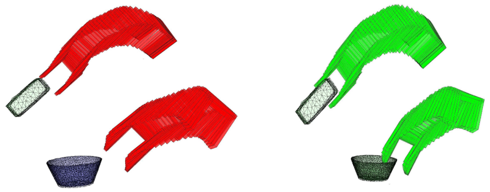

Fig. 6: Gripper trajectories for the Cereal (top) and Bowl (bottom) tasks. The trajectories on the left (red) were learned without grasp adaption, while the trajectories on the right (green) were produced with our approach.

Task \ Grasp	Cereal	Bowl	Door	Drawer	Slide	Average
Cereal	○	○	-	○	-	-
Bowl	○	○	-	○	-	○
Door	○	-	○	-	○	○
Drawer	○	○	-	○	-	○
Slide	-	-	-	-	○	-

TABLE I: Executing tasks with pre-defined grasps. Each column represents the grasp derived from a task or the average of all grasps. Rows show the tasks with ○ marking successes and - failures.

B. Grasp Adaption

The previous experiment demonstrated that an adaption from the human hand pose to the robot gripper is necessary. Here, we investigate in how far the automated grasp adaption from Sec. V-A can be replaced by a pre-given grasp offset. A reasonable assumption is that a grasp offset that works for one object might also work for another. As examples for working grasp offsets, we use the grasp offsets derived from the grasp adaption for each task. In addition, we also use the average of all grasp offsets. In comparison to computing $\boldsymbol{x}^*_g$, we transform the trajectories by fixed offsets, learn a model and execute that, i.e., we try each task with a grasp for each other task. Table I shows the results.

None of the grasps work for every object and there is also no task that can be solved with any grasp. The door and drawer tasks are solved with four of the six pre-given grasps as these have large handles (see Fig. 5(c) and (d)) and thus inaccurate grasps might still work. On the other hand the sliding door has a small handle (see Fig. 5(e)), which requires a specifically adapted grasp. This shows that a pre-

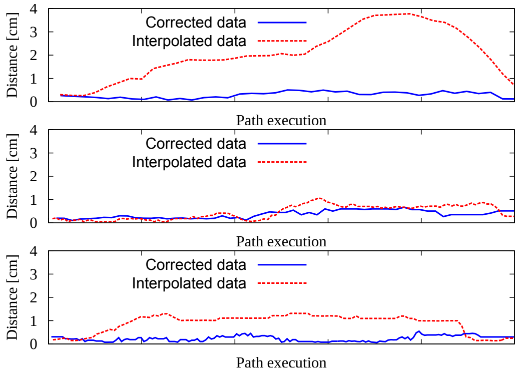

Fig. 7: Trajectory execution error for the Door (top), Drawer (middle) and Slide (bottom) tasks. The path execution has been scaled to the same length.

given grasp offset is unlikely to work in general and the grasp adaption ensures successful reproduction. This is due to the fact that the offset depends on the grasped object's shape, i.e., a wide grasp (cereals) leads to a different offset than a fingertip grasp (door/drawer handle). Further the marker placement on the hand underlies a small variation and the independent tracking of the hand and objects introduces an additional fluctuation. Note that our approach is represented by the diagonal entries as each grasp is adapted per object, however, in contrast to pre-given grasps automatically.

C. Trajectory Correction

In a third experiment we address the influence of the trajectory correction. We created motion models with grasp adaption either using the corrected trajectories $\hat{\boldsymbol{X}}_h$ and $\hat{\boldsymbol{X}}_o$ or the interpolated trajectories without correction $\boldsymbol{X}_h$ and $\boldsymbol{X}_o$ as input. As the grasp adaption prevents grasping failures, we now see the effect of the trajectory correction. We use the Door, Drawer and Slide tasks as these only allow constrained motions. Thus, during the execution the robot is forced away from the motion's trajectory. As we cannot measure these undesirable forces directly we consider the deviation of the actual execution from the learned motion. At each point of the execution, we compute the distance to the nearest point in the desired trajectory. Fig. 7 shows the results. With corrected trajectories, the execution error is always below one centimeter. The error is largest when using interpolated data for the Door task as linear interpolation does not accurately reflect the rotating motion. There is no clear difference for the Drawer task. The reason for that is that there is more mechanical compliance in the drawer mechanism and thus the robot could execute the motion without large deviations. This shows that the graph optimization is able to correct errors in the measurements and produce a smooth joint pose estimation for hand and object. Further it oppresses undesired (but present in the demonstrations) in-hand manipulations by fixing the grasp. It also fills occlusions in the demonstrations.

VII. Conclusion

We presented an approach that learns manipulation actions from human demonstrations. It aims towards a natural teaching process similar to how humans show tasks to others. In particular, we do not require continuous and highly accurate tracking, e.g., from a motion capture setup, data gloves, or kinesthetic teaching. Instead, our approach corrects gaps in trajectories using the background knowledge that manipulated objects move with the hand and by adapting demonstrated grasp trajectories to the robot's capabilities. In practical experiments we show that our approach is able to reproduce demonstrated object manipulation actions even when precise grasping is required. They further demonstrate that fixed grasp offsets are not sufficient in general.

References

[1] S. Calinon, Z. Li, T. Alizadeh, N. G. Tsagarakis, and D. G. Caldwell, "Statistical dynamical systems for skills acquisition in humanoids," in *Int. Conf. on Humanoid Robots (Humanoids)*, 2012.

[2] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal, "Learning and generalization of motor skills by learning from demonstration," in *Int. Conf. on Robotics & Automation (ICRA)*, 2009.

[3] S. Calinon, T. Alizadeh, and D. G. Caldwell, "On improving the extrapolation capability of task-parameterized movement models," in *Int. Conf. on Intelligent Robots and Systems (IROS)*, 2013.

[4] J. A. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *Int. Conf. on Robotics & Automation (ICRA)*, 2002.

[5] O. Kroemer, C. Daniel, G. Neumann, H. van Hoof, and J. Peters, "Towards learning hierarchical skills for multi-phase manipulation tasks," in *Int. Conf. on Robotics & Automation (ICRA)*, 2015.

[6] M. Mühlig, M. Gienger, and J. Steil, "Interactive imitation learning of object movement skills," *Autonomous Robots*, vol. 32, no. 2, pp. 97–114, 2012.

[7] T. Asfour, F. Gyarfas, P. Azad, and R. Dillmann, "Imitation learning of dual-arm manipulation tasks in humanoid robots," in *Int. Conf. on Humanoid Robots (Humanoids)*, Dec 2006, pp. 40–47.

[8] R. Lioutikov, G. Neumann, G. Maeda, and J. Peters, "Probabilistic segmentation applied to an assembly task," in *Int. Conf. on Humanoid Robots (Humanoids)*, 2015, pp. 533–540.

[9] A. ten Pas and R. Platt, "Using geometry to detect grasps in 3d point clouds," in *Int. Symposium of Robotics Research (ISRR)*, 2015.

[10] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic grasping of novel objects using vision," *Int. Journal of Robotics Research*, vol. 27, no. 2, pp. 157–173, 2008.

[11] A. T. Miller and P. K. Allen, "Graspit! – a versatile simulator for robotic grasping," *IEEE Robotics and Automation Magazine*, vol. 11, no. 4, pp. 110–122, 2004.

[12] G. Gioioso, G. Salvietti, M. Malvezzi, and D. Prattichizzo, "An object-based approach to map human hand synergies onto robotic hands with dissimilar kinematics," in *Robotics: Science and Systems (RSS)*, 2013.

[13] Y. Lin and Y. Sun, "Robot grasp planning based on demonstrated grasp strategies," *Int. Journal of Robotics Research*, vol. 34, no. 1, pp. 26–42, 2015.

[14] H. B. Amor, O. Kroemer, U. Hillenbrand, G. Neumann, and J. Peters, "Generalization of human grasping for multi-fingered robot hands," in *Int. Conf. on Intelligent Robots and Systems (IROS)*, 2012.

[15] K. Pauwels and D. Kragic, "Simtrack: A simulation-based framework for scalable real-time object pose detection and tracking," in *Int. Conf. on Intelligent Robots and Systems (IROS)*, 2015.

[16] M. Pardowitz, R. Haschke, J. Steil, and H. Ritter, "Gestalt-based action segmentation for robot task learning," in *Int. Conf. on Humanoid Robots (Humanoids)*, 2008, pp. 347–352.

[17] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *Int. Conf. on Robotics & Automation (ICRA)*, 2011.

[18] R. H. Byrd, J. C. Gilbert, and J. Nocedal, "A trust region method based on interior point techniques for nonlinear programming," *Mathematical Programming*, vol. 89, no. 1, pp. 149–185, 2000.

[19] K. Hsiao, S. Chitta, M. Ciocarlie, and E. G. Jones, "Contact-reactive grasping of objects with partial shape information," in *Int. Conf. on Intelligent Robots and Systems (IROS)*, 2010, pp. 1228–1235.

[20] S. Khansari-Zadeh and A. Billard, "Learning stable nonlinear dynamical systems with gaussian mixture models," *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 943–957, 2011.