Automated Robot Skill Learning from Demonstration for Various Robot Systems

Lisa Gutzeit[1⋆], Alexander Fabisch[2⋆], Christoph Petzoldt[1,3⋆⋆], Hendrik Wiese[2], Frank Kirchner[1,2]

[1]University of Bremen, Robotics Research Group, Germany
[2]German Research Center for Artificial Intelligence (DFKI GmbH), Robotics Innovation Center, Germany
[3]BIBA - Bremer Institut für Produktion und Logistik GmbH at the University of Bremen, Germany

Abstract

Transferring human movements to robotic systems is of high interest to equip the systems with new behaviors without expert knowledge. Typically, skills are often only learned for a very specific setup and a certain robot. We propose a modular framework to learn skills that is applicable on different robotic systems without adaptations. Our work builds on the recently introduced BesMan Learning Platform, which comprises the full workflow to transfer human demonstrations to a system, including automatized behavior segmentation, imitation learning, reinforcement learning for motion refinement, and methods to generalize to related tasks. For this paper, we extend this approach in order that different skills can be imitated by various systems in an automated fashion with a minimal amount of configuration, e.g., definition of the target system and environment. For this, we focus on the imitation of the demonstrated movements and show their transferability without movement refinement. We demonstrate the generality of the approach on a large dataset, consisting of about 700 throwing demonstrations. Nearly all of these human demonstrations are successfully transferred to four different robot target systems, namely Universal Robot's UR5 and UR10, KUKA LBR iiwa, and DFKI's robot COMPI. An analysis of the quality of the imitated movement on the real UR5 robot shows that useful throws can be executed on the system which can be used as starting points for further movement refinement.

1 Introduction and Related Work

Implementing new behaviors for robotic systems is tedious work. In recent years more and more often machine learning has been used to simplify this

[*]L. Gutzeit, A. Fabisch and C. Petzoldt have contributed equally as first authors.

problem. By using learning from demonstration techniques, intuitive knowledge from humans is leveraged to initialize a refinement process that has to be done on the real system. For this purpose we recently proposed the BesMan Learning Platform that automatizes the process of transferring relevant motion segments from human motion capture data to a robotic platform in a format that can be refined by standard policy search algorithms [7]. Most steps of this learning process, like the segmentation of the movement demonstration and the imitation learning, are already automatized. Manual work remains in defining a reward function to refine the skill.

In this work, we present a general approach to imitate human demonstrations on different robotic systems. To automatize this process for different robotic systems, poses of certain keypoints of the human body have to be mapped to poses of elements of the robotic system. This is a special case of the correspondence problem [14]. That means, demonstrated end effector trajectories have to be changed in a way that they are executable on the target system.

This process, which is also called motion retargeting, is a well-known problem in the computer graphics and animation community [5]. In practice it is often solved manually. In the context of imitation learning for robots it has been explored by [16, 12] with a fixed mapping. Similar work on automating the embodiment mapping has been published by [11]. Our approach to this problem has been presented briefly in [7] but not systematically analyzed so far. Our approach is more restricted: we do not integrate task-specific constraints in the optimization process (e.g., collision penalties for external objects, via-points) and we do not modify the shape of the trajectory. The benefit is twofold: our approach is more modular, which allows us to use any standard black-box optimization method to optimize the embodiment, and do task-specific adaptation with standard reinforcement learning algorithms. Furthermore, no model of the environment is needed at this stage. The only prior knowledge that is needed is a kinematic model of the robot. In this paper we particularly examine if this restricted approach already generates useful trajectories in the workspace of the robot. We build upon previous work [7] and evaluate the applicability of the learning platform to several different robotic target systems. The main focus lies on the automation of the embodiment mapping to map human movement recordings to a trajectory which is executable on the system.

In section 2, the general learning platform, including automatized movement segmentation and our approach to solve the correspondence problem in a general and easily configurable way, are described. In section 3, we present the experiments in which approx. 700 throwing motions have been recorded and transferred to the target system to evaluate the approach. The results of these experiments are presented and discussed in section 4 before we give a conclusion.

2 Generic Learning Platform

The BesMan Learning Platform is designed to cover the whole workflow of teaching robotic systems new behavior from human demonstrations. It provides

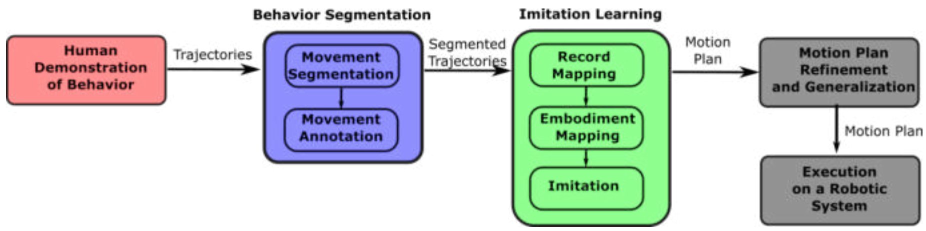


Figure 1: Overview of the learning platform. In this paper the focus is on the first three modules: recording of human demonstration and automatized movement segmentation and imitation.

utilities for human motion recording, behavior segmentation, imitation learning as well as motion refinement and generalization in a single framework [7].

In this paper we focus on the first part of the learning platform dealing with segmentation of human demonstrations and imitation learning. A general overview is shown in Figure 1.

Human demonstrations of the movements that will be transferred are recorded, automatically segmented into movement building blocks, and labeled according to known movement classes, as described in more detail in section 2.1. Afterwards, the segmented trajectories are mapped to the robot workspace and imitated, as described in section 2.2.

Due to the modular design of the approach, we can automatize the learning process for a new robotic system or environment by just requiring a minimal amount of configuration to integrate the available prior knowledge about the robot target system or the scenario respectively. Demonstrations of the same task can be transferred to different systems by configuring general robot related properties, like kinematic structure of the robot, type of gripper and default joint configuration (home position) and definition of a rotation matrix to map the robot's end effector rotation to the human hand orientation.

We summarize the methods used to automatically identify the relevant movement in the human demonstrations in the following section. Afterwards, methods for the transformation and optimization are presented, which are needed to make the movements executable on a robotic system. These methods are applicable on different robotic systems by just defining the robot configuration as described above.

2.1 Movement Segmentation

The learning platform allows for learning complex skills which cannot be learned monolithically. By splitting complex movements into simpler sub-tasks, learning of these tasks becomes feasible. Previously learned sub-tasks can be reused in other related tasks. In the learning platform, this process is automatized using the segmentation and classification approach presented in [8]. Human manipulation movements follow a characteristic bell-shaped velocity pattern of the hand [13]. We use these bell-shaped profiles to identify building blocks of manipulation movements using an unsupervised segmentation algorithm, called *velocity-based Multiple Change-point Inference (vMCI)*, introduced in [18]. To generate labels for the identified movement building blocks, we use a supervised approach based on 1-Nearest Neighbor (1-NN) classification. By transforming recorded trajectories of several positions of the demonstrator's arm to a coordinate frame relative to the back of the human, and interpolating the identified segments to the same length, simple 1-NN classification shows good classification results on several manipulation tasks with small training set sizes [8, 7]. Using the unsupervised movement segmentation algorithm vMCI in combination with 1-NN classification, the relevant movement parts of the demonstrations that shall be learned by the system can be selected. Benefits of the approach are that the movements can be performed naturally by the demonstrator, and no manual segmentation procedure is required. Furthermore, the same approach works for different manipulation movement without the need for adaptations. This has already been evaluated on ball-throwing and pick-and-place movements in our previous work [8, 6]. In this work, we additionally evaluate this approach on a bigger dataset, containing approximately 700 stick-throwing demonstrations.

2.2 Imitation Learning

It is not easily possible to directly transfer behaviors between humanoids (e.g. humans, humanoid robots, or robots which are similar to parts of humans) due to the correspondence problem [14]. To circumvent this problem, a record mapping is needed which maps marker trajectories of human demonstrations to a sequence of actions or system states, as well as an embodiment mapping, which maps the recorded sequence to a trajectory that is executable on the target system [1]. In this section we present solutions for the record and embodiment mapping as well as an optimization of the embodiment mapping which is applicable on different robotic systems.

Record Mapping

We assume that the demonstrator's state-action sequence can be discretized with

$$\boldsymbol{\tau}^D = \left(\boldsymbol{s}_0^D, \boldsymbol{a}_0^D, \ldots, \boldsymbol{s}_{T-1}^D, \boldsymbol{a}_{T-1}^D, \boldsymbol{s}_T^D\right),$$

and that there is some underlying policy that has been used to generate $\boldsymbol{\tau}^D$, which is fully defined by a probability density function $\pi^D(s, a)$. $\boldsymbol{\tau}^D$ cannot be directly observed because neither the actions $\boldsymbol{a}_t^D, t \in \{0, \ldots, T-1\}$ (e.g., muscle contractions of a human) can be observed directly, nor can the states $\boldsymbol{s}_t^D, t \in \{0, \ldots, T\}$ be observed in their entirety (e.g. configurations of all joints of a human). Instead, $g^R(\boldsymbol{s}^D, \boldsymbol{a}^D) = \left(\boldsymbol{s}^R, \boldsymbol{a}^R\right)$ can be observed and recorded, where information can be lost (e.g., some joint configurations, muscle states, etc.) in the *record mapping* $g^R : S^D \times A^D \rightarrow S^R \times A^R$. In our case, marker positions on the human's body are captured by a motion capture system and represented in the motion capture system's world coordinate system. The information about the state of the human is limited by the amount of markers that are attached to the body, and we cannot measure joint angles directly because we attach markers exclusively to the skin and clothes of the human, hence poses can change slightly over time without any actual joint movement. From these markers, hand pose trajectories can be extracted and represented relative to the pose of the human's back.[1] Our setup allows to easily add tracking of dual-hand motions as well. Hence, we extract a sequence $\boldsymbol{\tau}^R$ that contains poses of end effectors.

Embodiment Mapping

We assume that in the ideal case all states and actions can be recorded perfectly; that is, g^R is the identity. We would like to infer the policy π^D based on the collected data $\boldsymbol{\tau}^R = \left(\boldsymbol{s}_0^R, \boldsymbol{a}_0^R, \ldots, \boldsymbol{s}_{T-1}^R, \boldsymbol{a}_{T-1}^R, \boldsymbol{s}_T^R\right)$. However, it is often not possible to reach all states or execute all actions. For example, if a movement from a human shall be transferred to a humanoid robot, it might be the case that the robot has not as many degrees of freedom as the human and thus cannot execute a similar, smooth trajectory within the workspace. It might be impossible to reach several states because they are not in the robot's workspace, the robot might be too weak or slow to execute the desired motions, or it might be to stiff and heavy to generate the same sequence of states. Actions are ignored for this discussion. They can be generated by position or velocity controllers on the real system. That means, an *embodiment mapping* $g^E : S^R \times A^R \rightarrow S^E \times A^E$ is required, which maps the recorded data to states and actions that are executable on the target system. g^E is not given like g^R, but instead has to be constructed either manually, from data, or both. As in Section 2.3.2 of [7], we propose to use simple black-box optimization to find g^E.

Optimization of Embodiment Mapping

In our work, we observe sequences of end effector poses (trajectories) from, e.g., a human teacher. Thus, an embodiment mapping has to be obtained that maps these trajectories to the workspace of the robot such that they are reachable and there are no discontinuities in joint space. We propose a parameterized linear mapping of the form

$$g^E(x_t, y_t, z_t, \alpha_t, \beta_t, \gamma_t) = \begin{pmatrix} \boldsymbol{R}_{\alpha,\beta,\gamma} \left((1-s) \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} + s \begin{pmatrix} x_t \\ y_t \\ z_t \end{pmatrix} \right) \\ \alpha + \alpha_t \\ \beta + \beta_t \\ \gamma + \gamma_t \end{pmatrix} + \boldsymbol{b},$$

where $s \in [0, 1]$ is a scaling factor, $\boldsymbol{\theta} = \alpha, \beta, \gamma$ are Euler angles (rotation around x-, y'- and z"-axis) that define a rotation, and $\boldsymbol{b}$ is an offset vector. $s, \boldsymbol{\theta}, \boldsymbol{b}$ will

[1] We use `pytransform3d` to calculate these transformations [3].

be selected to maximize the objective

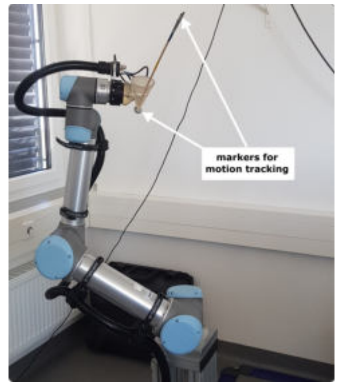
$$
\begin{aligned}
f(s, \boldsymbol{\theta}, \boldsymbol{b}) \quad = \quad & \exp\left(\frac{10}{T+1} \sum_t r(g^E(\boldsymbol{p}_t))\right) \\
& - w_{vel} \sum_t \dot{\boldsymbol{q}}(g^E(\boldsymbol{p}_t)) - w_{acc} \sum_t \ddot{\boldsymbol{q}}(g^E(\boldsymbol{p}_t)) - w_{jrk} \sum_t \dddot{\boldsymbol{q}}(g^E(\boldsymbol{p}_t)) \\
& - w_{coll} \sum_t c(\boldsymbol{q}_t) - w_{dist} ||\boldsymbol{p}_T|| \\
& + w_{height} \sum_t \boldsymbol{p}_{3,t} + w_{size} \sum_{d=1}^{3} \max_t \boldsymbol{p}_{d,t} - \min_t \boldsymbol{p}_{d,t},
\end{aligned}
$$

where $t \in \{0, \dots, T\}$ is the time step, $r(\boldsymbol{p})$ is 1 if $\boldsymbol{p}$ is a reachable end effector pose and 0 otherwise, $c(\boldsymbol{q})$ is 1 if the configuration results in self-collision and 0 otherwise, $\boldsymbol{p}_t$ is an end effector pose and $\boldsymbol{q}_t$ are corresponding joint angles at step t. The objective maximizes reachability, while minimizing the risk of getting too close to singularities, avoiding self-collisions, and maximizing exploitation of the robot's workspace. To maximize f any black-box optimizer can be used. We decided to use covariance matrix adaptation evolution strategies (CMA-ES; [9]) for this paper. The weights have to be configured appropriately. Depending on the target system, an analytic solution to inverse kinematics, a numerical solution, or even an approximation of a numerical solution [4] can be used if it is difficult to find a mapping that fits the trajectory into the workspace of the robot. In this paper we use both a numerical solution and an approximation, and take the solution that yields the best result. In addition, the resulting trajectory is smoothed with a mean filter in Cartesian space and in joint space (positions and accelerations) to avoid infeasibly high accelerations.

Imitation Learning

After mapping the recorded trajectory to the robot's workspace, a suitable representation that can be used for further adaptation and refinement is needed. A popular class of policy representations that has been used to learn movement primitives for robotic manipulation are Dynamical Movement Primitives (DMPs; [10, 15]). There are numerous advantages of DMPs in comparison to other policy representations for our purposes, among them: (1) They are stable trajectory representations. Slight errors in execution of the trajectory will not result in error accumulation like in general function approximators. (2) To reproduce a demonstrated movement, a one-shot learning algorithm can be used that determines the weights of the forcing term θ_i. Hence, imitation learning with DMPs is much simpler than it is for more general function approximators. (3) Movements can be easily adapted (even during execution): the goal of the movement can be changed and obstacles can be avoided.

(a) Setup to record the human arm motion with markers attached to the back, arm and hand.



(b) To record the stick position, a marker is attached to the tip of the stick. The movement of the robotic arm UR5 is tracked with a marker attached to the end effector

Figure 2: Motion recording setup.

3 Experiments

In this section, we evaluate the proposed configurable learning platform on a Touhu scenario.

Touhu, also known as *pitch-pot*, is a throwing game that is traditionally played in Eastern Asia. The goal is to throw a stick from a given distance into a pot. We use this scenario to evaluate the transferability of throwing movements to different robotic systems using their kinematic models.

Additionally, the quality of the transferred movements is evaluated in a second experiment, where the trajectory and goal position of the stick thrown by humans is compared to the one thrown by a real UR5 robot.

3.1 Experimental Setup

The throwing motions demonstrated by the human subject are recorded with a marker based-motion capturing system. The position of infrared light reflecting markers attached to the human body are recorded by several cameras at 500 Hz. To record the throwing motions, markers are attached to the hand, elbow, shoulder and back of the subject to track these positions. To determine the record mapping as described in section 2.2, additionally to the position of the back, its orientation is also needed. Thus, we attached three markers to the back instead of one, relative to each other in a well-known setup. The setup allows for determining the orientation of the markers. The same applies to the hand to allow for tracking the hand pose including its orientation. The complete marker setup can be seen in fig. 2.

In the second experiment, in addition to the human movement, likewise the position of the stick is of interest. With it, the trajectories of the thrown stick from the demonstration can be compared to the resulting trajectories after imitation of the throwing motion on the real system. Thus, a maker is placed on one end of the stick. Additionally, the movement of the robotic arm UR5 is captured by placing a marker at the end effector (see again fig. 2). The recorded demonstrations were down-sampled to 30 Hz and automatically segmented into movement building blocks with a bell-shaped velocity profile of the hand using the vMCI algorithm as described in section 2.1. Using 1-NN classification, the movement segments were classified into the classes *strike_out*, *throw*, *swing_out* and *idle*. To transfer the recorded demonstrations to the four robotic systems, the embodiment mapping has been optimized with the following weights in the objective function: $w_{coll} = 100, w_{vel} = 100, w_{acc} = 10, w_{jrk} = 1, w_{dist} = 0.1, w_{height} = 50, w_{size} = 100$. These weights have been determined empirically. The optimization was limited to the Cartesian translation of the trajectory within the workspace of the robot. Orientation and scaling remained unchanged.

3.2 Transfer in Simulation

Throwing motions of seven different subjects, each performing between 41 and 246 throws, were recorded to evaluate the generality of our approach. In total, 697 Touhu demonstration were recorded. With this large dataset, we are able to evaluate the generality of the movement segmentation as well as the transfer to different robotic systems, namely Universal Robots' UR5 and UR10, KUKA LBR iiwa and DFKI's COMPI [2]. To evaluate the movement classification, a stratified cross-validation repeated 100 times was performed with a fixed number of examples per class in the training data. Segments which could not be clearly assigned to one of the movement classes were removed from this evaluation. Furthermore, the number of successfully transferred movements as well as the difference between the position of the hand in the demonstrations and the end effector position of the systems are analyzed.

3.3 Transfer to a Real System

In the second part of the Touhu experiment, we additionally recorded 34 throwing movements of three subjects, in which also the position of the stick was recorded. Three subjects performed 10, 11 and 13 throws respectively. These demonstrations were transferred to the real UR5 robotic arm. We analyzed the number of successful throws, the stick position during the throw and its goal position. The transferability of the demonstrated throws on the UR5 robot is evaluated with respect to the following aspects: Does the robot inadvertently collide with anything including the stick? Does the stick fall out of the stick holder while the robot approaches the starting pose of the trajectory? Does the stick leave the holder during the throwing motion? If any of these aspects are evaluated negatively, the trajectory is considered not transferable. To evaluate the quality of the transferred throws, we compare the stick trajectories of the demonstrated throws and the recordings of the throws transferred to the UR5. Since the motion capture system sometimes returned noisy stick position measurements, the trajectories had to be interpolated. A quadratic model was used for interpolation. The same model has been used to extrapolate both the demonstrated and the reproduced stick trajectory until the stick hit the ground. Before this, we aligned the demonstrated trajectory with the start position of the transferred one. Thus, the distance between ground contact points as well as the similarity of the the stick trajectories can be determined. We use the average dynamic type warping (DTW; [17]) distance, i.e., the DTW distance divided by the maximum number of steps of the two time series, to compare the trajectories.

4 Results

4.1 Transfer in Simulation

The automatic segmentation of 697 demonstrated throws resulted in 2913 identified segments with a bell-shaped velocity profile. Although some throwing demonstrations showed just a small decrease of the velocity of the hand between throw and swing out phase of the movement, most of the throwing segments were successfully segmented. As an example result, a demonstration of one subject containing 41 throws is visualized in Figure 3.

2233 segments were used to evaluate the annotation. The number of training examples per class was varied between 1 and 20. With 4 examples per class a mean classification accuracy of 90 % could be achieved. 95 % could be achieved with 9 examples per class. Thus, a training data set with 9 examples per class was created, which contains the first three throwing demonstration of three different subjects. Using this training data, the segments of all recorded demonstration were classified. The movement class *throw* could be detected with an accuracy of 99 %, with 623 correctly detected throwing movements, 13 false negatives and 2 false positives. In Figure 3, different colors indicate the resulting labels of subject 5. The segments of this subject were classified with an accuracy of 98 %, with 24 segments that had to be removed from evaluation. This result shows that the approach to identify the relevant parts in the demonstrations also generalizes to larger datasets. A small number of labeled training data was sufficient to annotate the automatically derived segments. The throwing trajectories mapped into the workspace of the robotic system UR5, UR10, KUKA LBR iiwa and COMPI, are shown in Figures 4a and 4b. 682 trajectories were transferred to the workspaces of all target systems. We can see that most trajectories easily fit in the workspace of UR10 (arm radius: 1300 mm) and KUKA LBR iiwa 7 (arm radius: 1266 mm), while many trajectories have to be distorted or are close to the borders of the workspace of UR5 (arm radius: 850 mm) and COMPI (arm radius: 940 mm). Throwing movements often tend to be close to the borders of the human's workspace. Hence, the skill that we selected is quite challenging for smaller robots. The middle row of Figure 4

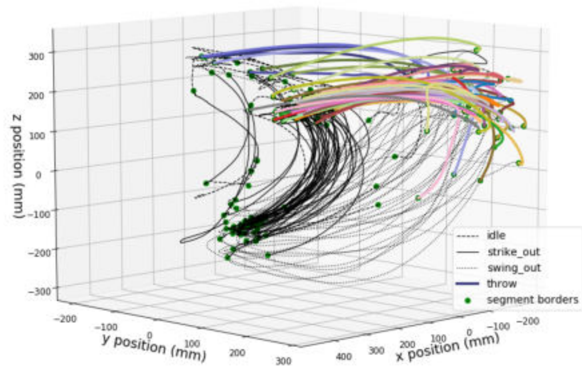


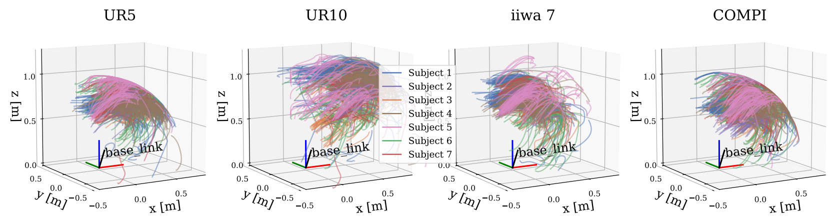
Figure 3: **Throwing trajectories of subject 5.** The position of the hand is shown for one set consisting of 41 Touhu-throws. Green dots mark the result of the vMCI segmentation. The resulting segment trajectories were labeled with 1-NN classification. Different line styles mark the different classes. Throwing segments are visualized as straight lines with a different color for each throw.

shows the demonstration of subject 5 visualized in Figure 3 transferred to the robotic arms. The different colors match the colors of the throwing segments in Figure 3.

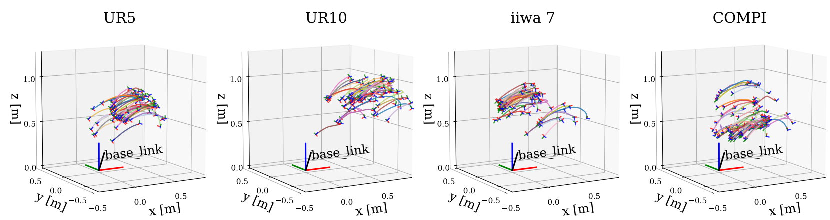
Figure 4c shows the ground contact points of sticks for the presented throwing trajectories from simulation. It can be seen that on average the UR10 has the widest distribution as it has the largest workspace. We quantify how well the throwing trajectories can be transferred to the real UR5 robot, as it is one of the more challenging robotic systems due to the more restricted workspace.

4.2 Transfer to the real system

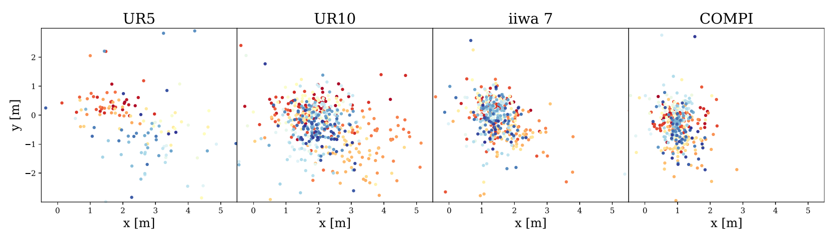
In this experiment, the throwing motions have been detected with an accuracy of 97 %, using the same training data as in the first experiment. 33 throws were correctly detected and one was wrongly assigned to another class by the 1-NN classification. 27 out of these segments could be transferred to the real UR5. A comparison of the stick trajectories can be seen in Figure 5a, in which the best, a good and the worst result in visualized. The mean average DTW distance is 0.15 m (standard deviation: 0.1 m), and the mean goal distance is 0.72 m (standard deviation: 0.31 m). The full error distribution is shown in Figure 5. The results show, that it is possible to automatically imitate demonstrated throws and that most of these throws are executable on the real system. Furthermore, the executed movements show useful throws. However, the goal positions of the



(a) All recognized throwing movements transferred to the workspace of the four robots. Trajectories from the same subject are shown in the same color



(b) All throwing movements of subject 5. Colors indicate the indices of the throws and correspond to the colors in Figure 3. The frames of the robots' base links are shown.



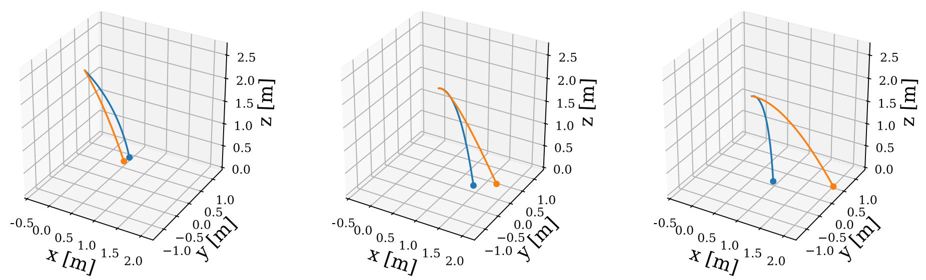
(c) Throwing results in simulation. We display the distribution of ground contact points of the sticks. Colors of the points indicate the index of the transferred throwing trajectory

Figure 4: **End-effector trajectories of throwing movements in robots' workspaces and corresponding ground contact points of the sticks.**

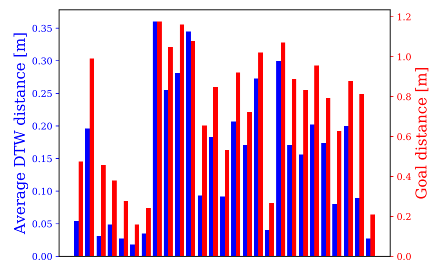
demonstrated throws are not reached by the system.

5 Conclusion

This work is built on our previous work about the platform for learning robot skills from humans, presented in [7]. We extend this work to a more general approach applicable to several systems with little configuration overhead. Throwing trajectories are automatically extracted from human demonstrations, and transferred to four robotic target systems. We show that the embodiment mapping, which is needed to map human movement trajectories into the robot workspace, can be automized for a dataset of 697 throws. Throwing is a chal-



(a) 3D plot of stick trajectories of the best (left), a good (middle), and the worst (right) result. The orange trajectory indicates how the stick was thrown in the demonstration and the blue trajectory is the reproduction by the UR5



(b) Average dynamic time warping distances and distances of goal positions.

Figure 5: Analysis of the execution of throws on the real UR5

lenging skill for these robots because it has high acceleration and velocities and is close to the border of the workspace of humans. Nonetheless, most of the demonstrated throws could be transferred to the systems using our approach. Furthermore, we evaluate the difference of stick trajectories and ground contact points between demonstrated throws and reproductions of those on a real UR5. We show that there is still a significant gap between the outcome of demonstrated throws and their reproductions. As proposed earlier [7], a solution to this problem would be to use reinforcement learning to refine the motion.

Acknowledgements

This work was supported through grants from the German Federal Ministry for Economic Affairs and Energy (BMWi, No 50RA1703, No 50RA1701),one grant from the European Union's Horizon 2020 research and innovation program (No H2020-FOF 2016 723853), and part of the work was done in a collaboration with Intel Labs China. We would like to thank Intel Corp. for financial support.

References

[1] Argall, B.D., Chernova, S., Veloso, M., Browning, B.: A survey of robot learning from demonstration. Robotics and Autonomous Systems **57**(5), 469–483 (2009)

[2] Bargsten, V., de Gea Fernández, J.: Compi: Development of a 6-dof compliant robot arm for human-robot cooperation. In: Proceedings of the 8th International Workshop on Human-Friendly Robotics. Technische Universitaet Muenchen (TUM) (2015)

[3] Fabisch, A.: pytransform3d: 3d transformations for python. Journal of Open Source Software **4**. https://doi.org/10.21105/joss.01159

[4] Fabisch, A.: A comparison of policy search in joint space and cartesian space for refinement of skills. In: Advances in Intelligent Systems and Computing. vol. 980. Springer (2019), accepted

[5] Gleicher, M.: Retargetting motion to new characters. In: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques. pp. 33–42. SIGGRAPH '98, ACM, New York, NY, USA (1998). https://doi.org/10.1145/280814.280820, `http://doi.acm.org/10.1145/280814.280820`

[6] Gutzeit, L., Otto, M., Kirchner, E.A.: Simple and robust automatic detection and recognition of human movement patterns in tasks of different complexity. In: Physiological Computing Systems. Springer (2019), submitted

[7] Gutzeit, L., Fabisch, A., Otto, M., Metzen, J.H., Hansen, J., Kirchner, F., Kirchner, E.A.: The BesMan Learning Platform for Automated Robot Skill Learning. Frontiers in Robotics and AI **5** (may 2018). https://doi.org/10.3389/frobt.2018.00043, `https://www.frontiersin.org/articles/10.3389/frobt.2018.00043/full`

[8] Gutzeit, L., Kirchner, E.A.: Automatic detection and recognition of human movement patterns in manipulation tasks. In: Proceedings of the 3rd International Conference on Physiological Computing Systems (2016)

[9] Hansen, N., Ostermeier, A.: Completely derandomized self-adaptation in evolution strategies. Evolutionary Computation **9**, 159–195 (2001)

[10] Ijspeert, A.J., Nakanishi, J., Hoffmann, H., Pastor, P., Schaal, S.: Dynamical movement primitives: learning attractor models for motor behaviors. Neural Computation **25**(2), 328–373 (2013)

[11] Maeda, G.J., Ewerton, M., Koert, D., Peters, J.: Acquiring and Generalizing the Embodiment Mapping From Human Observations to Robot Skills. IEEE Robotics and Automation Letters (RA-L) **1**(2), 784–791 (2016). https://doi.org/10.1109/LRA.2016.2525038

[12] Michieletto, S., Chessa, N., Menegatti, E.: Learning how to approach industrial robot tasks from natural demonstrations. In: 2013 IEEE Workshop on Advanced Robotics and its Social Impacts. pp. 255–260 (Nov 2013). https://doi.org/10.1109/ARSO.2013.6705538

[13] Morasso, P.: Spatial control of arm movements. Experimental Brain Research **42**, 223–227 (1981)

[14] Nehaniv, C.L., Dautenhahn, K.: Imitation in animals and artifacts. chap. The Correspondence Problem, pp. 41–61. MIT Press, Cambridge, MA, USA (2002)

[15] Pastor, P., Hoffmann, H., Asfour, T., Schaal, S.: Learning and generalization of motor skills by learning from demonstration. In: IEEE International Conference on Robotics and Automation. pp. 763–768 (2009)

[16] Pollard, N.S., Hodgins, J.K., Riley, M.J., Atkeson, C.G.: Adapting human motion for the control of a humanoid robot. In: Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292). vol. 2, pp. 1390–1397 vol.2 (May 2002). https://doi.org/10.1109/ROBOT.2002.1014737

[17] Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. IEEE Transactions on Acoustics, Speech, and Signal Processing **26**(1), 43–49 (February 1978). https://doi.org/10.1109/TASSP.1978.1163055

[18] Senger, L., Schröer, M., Metzen, J.H., Kirchner, E.A.: Velocity-based multiple change-point inference for unsupervised segmentation of human movement behavior. In: Proccedings of the 22th International Conference on Pattern Recognition (ICPR2014). pp. 4564–4569 (2014). https://doi.org/10.1109/ICPR.2014.781