One-Shot Robot Programming by Demonstration using an Online Oriented Particles Simulation

Christian Groth and Dominik Henrich[†]
Lehrstuhl für Angewandte Informatik III
Universität Bayreuth, D-95445 Bayreuth, Germany

***Abstract*— To provide robots for a wide range of users, there needs to be an easy and intuitive way to program them. This issue is addressed by the robot programming by demonstration paradigm, where the user demonstrates the task to the robot. While there exist a lot approaches that use multiple demonstrations for the learning procedure, single-shot robot programming by demonstration is still a niche. Also, all available approaches in this niche have severe drawbacks.**
The main contribution of this work is a novel one-shot programming by demonstration approach, that performs an online adaption of a provided trajectory to a new situation. For that, the system regards every sample of the trajectory and every reference (object) in the scene as a particle. These particles interfere with each other by forces and torques that arise from inherent potential fields. Thus, in a new situation the trajectory will adapt to the potential fields of the relocated references and converge to a minimum energy state. We evaluated the approach qualitatively and quantitatively using cross validation.

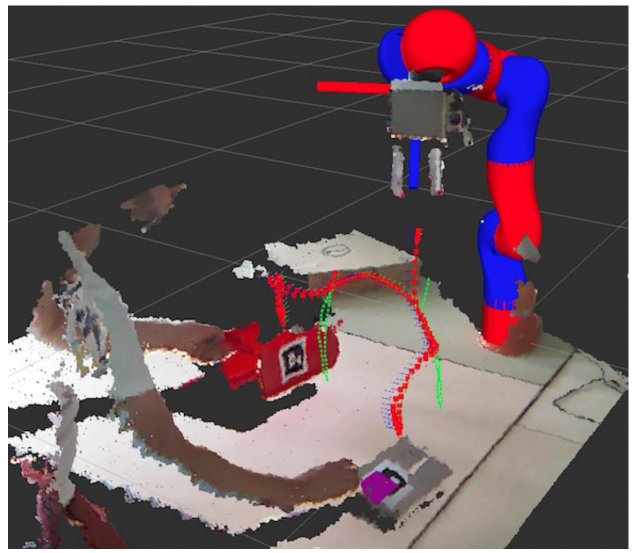


Fig. 1. Online adaption of a demonstrated trajectory using oriented particles. The original demonstration (green) and the adaption (red) is projected into the RGB-D image of the scene. Also, the robot is overlaid with its model for better visualization.

I. Introduction and State of the Art

Robots should do useful work and serve people. In order to reach this goal still a lot of issues need to be addressed. One of these issues is the easy and intuitive programming of robots. Since traditional robot programming is only feasible for experts, there needs to be a simple way to program new abilities to a robot. Programming by demonstration, also known as Learning From Demonstration or Imitation Learning is one such approach, where the user can teach new skills to the robot by providing one or more demonstrations of the task [1]. The system will extract relevant features and generate a generalized version of the task, which allows the robot to reproduce the task in a new situation.
If the user provides a set of demonstrations, machine learning algorithms can be applied to extract relevant features from the demonstrations. The most promising and widely used approach is the parameter estimation of Gaussian mixtures. By providing demonstrations with altering variance over time, the system can distinct between relevant and irrelevant parts. The reproduction is achieved through statistical regression techniques. The approach can be supplemented in many directions. Recent research includes subtask extraction [2], the combination with dynamic movement primitives [3], incremental online learning of the mixtures [4], the incorporation of redundancy [5], the integration of forces by adding virtual springs [6] and the augmentation with an obstacle avoidance [7]. All of the approaches demand from the user to provide a set of distinct demonstration, so the system can recognize relevant parts indicated by a low variance. If the number of demonstrations is not sufficient, the algorithms are not able to generalize.
To facilitate the programming of the robot, many approaches try to lower the number of demonstrations. A widespread approach is the use of so called (dynamic) movement primitives or action primitives. Here, the programming is carried out as a two-step process. In a first step, action primitives are demonstrated, which act as building blocks. In a second step, the user demonstrates a complete task, which allows the system to identify the required primitives. There exist approaches to generate dynamic movement primitives from a single demonstration [8] [9] [10]. But only few work has been done on coordinating these primitives in order to recognize the complete task [11] [12] [13] [14].
A niche in the field of programming by demonstration is one-shot learning, where the robot adapts a complete task to a new situation from a single demonstration. One of the previously proposed approaches is only suitable for mobile

[†]This work has partly been supported by the Deutsche Forschungsgemeinschaft (DFG) under grant agreement He2696/15 INTROP.

robots [15]. The adaption of the trajectory using fluid dynamics is only capable of two-dimensional trajectories and requires manual post-processing [16].
Also, a global warping mechanism can be used to adapt a trajectory [17]. But it is computationally expensive ($O(n^3)$) and cannot handle the orientation of the robot. The approach in [18] uses a so called non-rigid registration to transfer a trajectory. But the authors state, that the approach relies on strong assumptions about the registration and may fail in complicated situations. Although there exist promising ideas, none of the approach satisfies our demands.
In order keep the work for the user as small as possible, we will investigate if we can teach the robot useful tasks with just a single demonstration. The authors in [19] state, that generalizing observations into a set of intrinsic parameters limits the ability of user interaction, therefore we will concentrate on the trajectory level.
The contribution of our work is a new approach to adapt online a single user-provided demonstration online to a new situation. In contrast to existing work in this niche, neither an existing library of primitives nor manual post processing is required and the algorithm is capable of adapting not only the position of the robots tool center point, but also the orientation in 3D space to the new scene. Furthermore our approach provides a natural integration of a collision avoidance, to adapt the trajectory in dynamic environments.

II. Problem Formulation

The goal of the approach is to enable the robot to execute an adapted version of a task in a new situation. We regard a new situation as changed object positions and orientations and a changed starting pose of the robot. The user provides just one demonstration of the task. The approach should adapt the demonstration to the new situation. The robot should not only fulfill the task in the new situation but also the generated motion should be similar to the demonstration. Furthermore the reproduced motion should be identical to a demonstration under unchanged conditions, since it reflects the ground truth provided by the user.
In this work, we regard robotic manipulators mounted e.g. on a table with a certain range to operate. A demonstration consists of a trajectory T and a set of 6-dimensional references R. The trajectory $T = \{t_0, \ldots, t_{n-1}\}$ represents the $n \in \mathbb{N}$ samples(via-points) of the robot's tool center point t_i in homogeneous 3D coordinates, including position and orientation. A reference r_i of $R = \{r_0, \ldots, r_{m-1}\}$ with $m \in \mathbb{N}$ represents the position and orientation of e.g. an object or a marker in homogeneous 3D coordinates. A sample t_i carries additional information, like gripper operations. A reference r_i carries additional features for identification.
In order to adapt a demonstration to a new situation, we need a matching of every reference $r_i \in R^D$ of the demonstration to the set of available references R in the current scene. This can be achieved by feature matching or an object recognition. The trajectory should then be bent accordingly to the new positions and orientations of the references in the environment. Additionally the bent trajectory should not collide with obstacles in the new situation.

III. Approach

In this section we will describe our particle-based approach. In brief, we regard the user-demonstrated trajectory as a set of particles. Each of these particles has an inherent potential field. This potential field applies forces and torques only on linked particles. To model the influence of the environment, we also consider objects (references) as particles, which influence the particles of the trajectory (Figure 2). We

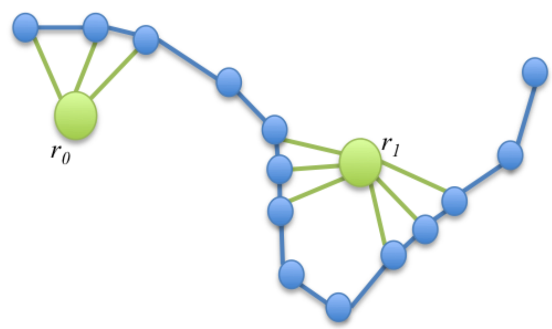


Fig. 2. Trajectory as particle chain (blue), partially linked to objects r_i represented by particles (green), too.

consider the provided demonstration as an equilibrium of all particles. When the demonstration is adapted to a new situation, the changed object positios will apply forces and torques to the particles since the equilibrium is disturbed. The trajectory of particles will adapt to the new situation by converging to a minimum energy state, which will also maintain the overall shape of the trajectory. A collision avoidance is easily integrated by adding repulsive potentials to obstacles each represented as one or multiple particles.

A. Particle Dynamics

As stated, we consider every sample of the demonstrated trajectory and each object as an oriented particle [20]. For every trajectory particle t_i we apply the standard Newtonian equations of motion

$$a_i = \frac{F_i}{m_i} \qquad \alpha_i = I_i^{-1}\tau_i$$

$$v_i = \dot{a}_i \qquad \omega_i = \dot{\alpha}_i$$

$$p_i = \dot{v}_i \qquad q_i = \dot{\omega}_i$$

where p_i is the position and q_i is the orientation of the particle. The m_i is the particles mass, I_i is the rotational inertia, a_i and v_i are the acceleration and velocity, and α_i and ω_i are the angular acceleration and velocity.
We calculate the forces F_i and torques τ_i for each particle i as:

$$F_i = \sum_{j \neq i} F_{s,i}(j) + \sum_{r \in R} F_{r,i}(r) + \sum_{o \in R} F_{c,i}(o) - F_{VD}$$

$$\tau_i = \sum_{j \neq i} \tau_{s,i}(j) + \sum_{r \in R} \tau_{r,i}(r) - \tau_{VD}$$

where $F_{r,i}$ and $\tau_{r,i}$ arise from the changed constraints of the scene and its references R. The $F_{s,i}$ and $\tau_{s,i}$ denote forces and torques, that arise from the trajectory particles. Additionally we add damping terms F_{VD} and τ_{VD} and a collision avoidance force $F_{c,i}$.

B. Force And Torque Calculation

The forces and torques are calculated upon different potentials. Each particle inherits a shape potential ϕ_s, that tries to replicate the shape of the trajectory. In other words, a particle tries to remain its position and orientation from the demonstration w.r.t. its neighbors.

Thereto, we analyze the interconnections of the particle t_i to its neighbors t_j. We re-project every particle t_j as t_i^j, based on the initially demonstrated relationship t_i^D and t_j^D.

$$t_i^j = t_j \cdot t_j^{\mathrm{D}^{-1}} t_i^{\mathrm{D}}$$

We denote p as the position of t and q as the orientation of t in quaternion notation.

The resulting force $F_{s,i}$ on particle p_i is calculated as

$$F_{s,i}(p_j) = -\nabla_p \phi_s = |p_i^j - p_i|^2 \cdot (p_i^j - p_i).$$

The torque is calculated as

$$\tau_{s,i}(q_j) = -\nabla_q \phi_s = |q_i^j \cdot q_i^{-1}|^2 \cdot (q_i^j \cdot q_i^{-1}).$$

The force and torque will push the particle t_i back to its relative position and orientation w.r.t. particle t_j from the demonstration. The force and torque will increase with growing displacement between demonstrated and current position. This will result in a global maintaining of the trajectory.

To adapt the trajectory to the constraints of the current situation, a reference potential ϕ_r is applied. It tries to replicate the trajectory w.r.t. each reference within the situation. Therefore the force $F_{r,i}$ and the torque $\tau_{r,i}$ are applied to the trajectory particles. We calculate the forces analogously, but for all linked reference particles t_r.

$$t_i^r = t_r \cdot t_r^{D^{-1}} t_i^D$$

$$F_{r,i}(p_r) = -\nabla_p \phi_r = |p_i^r - p_i|^2 \cdot (p_i^r - p_i) \cdot \psi(p_i^D, p_r^D)$$

Analogously for the torque:

$$\tau_{r,i}(q_r) = -\nabla_q \phi_r = |q_i^r \cdot q_i^{-1}|^2 \cdot (q_i^r \cdot q_i^{-1}) \cdot \psi(p_i^D, p_r^D)$$

Where ψ is a weighting term, calculated as:

$$\psi(p_i^D, p_j^D) = \frac{d_{max} - |p_i^D - p_j^D|}{d_{max}}$$

In our case, we want to make sure that positions of the tools center point w.r.t. objects are reproduced more exact with lower distance. I.e. contact states must be identical, while approaching motions may vary with increasing distance. The estimation of parameter d_{max} will be discussed in the next section.

A collision potential ϕ_c shall avoid collisions of particles with obstacles o in the scene. Since we only want to avoid the collision, we do not need to alter the orientation. We calculate the force $F_{c,i}$ between particle position p_i and obstacle position p_o as

$$F_{c,i} = -\nabla \phi_c = \frac{1}{|p_i - p_o|^2} \cdot (p_i - p_o)$$

We also apply a damping realized through viscous drag to avoid high translation and angular velocities. The damping force and torque are calculated as

$$F_{VD} = -v \cdot s_{VD}$$

$$\tau_{VD} = -\omega \cdot s_{VD}$$

with a scalar damping factor s_{VD}.

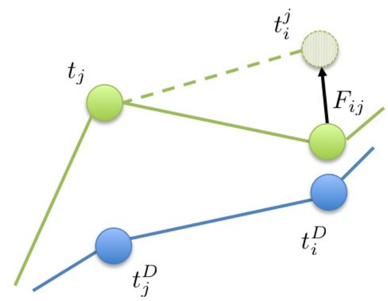


Fig. 3. Force $F_{s,i}(j)$, denoted as F_{ij} resulting from the re-projection of t_i^D w.r.t. t_j as t_i^j

C. Particle Links

In particle simulations usually all particles within a certain range influence each other. I.e. the distances between all particles are calculated and scaled forces are applied to each particle. This often leads to a run-time complexity of $O(n^2)$. Since in our case the particles have a strict temporal order, we do not need to calculate all possible combinations. We only calculate and apply the forces and torques to linked particles. We choose links complying with our domain and link particle t_i with its succeeding particle t_{i+1}. This will attempt to maintain the overall shape of the trajectory. We also link some of the trajectory particles with reference particles. This refers directly to "what has to be learned" of the five-W problems from Section I. In the machine learning approaches, the problem is usually solved implicitly. An approach, that deals particularly with the problem can be found in [2]. Since it is not the focus of this paper, we will use the regular way of particle simulation and link all trajectory particles within a certain range d_{max} to the object particle. This is valid since in pick-and-place and manipulation tasks, different parts of a demonstration usually relate to different objects. We set $d_{max} = 2 \cdot s_{max}$, where s_{max} is the largest diameter of each of the reference objects bounding spheres. We use this high threshold because of the possible usage of an object as a tool. More domain-specific approaches could be exposed in future work.

We also link all trajectory particles dynamically to obstacles in the scene to apply the collision avoidance force. As an obstacle, we regard every reference in the scene, which was not linked to the particle due to the demonstration. A link for collision avoidance is created, if a particle's distance is lower than twice of the obstacle's bounding sphere. The link is removed again, if the distance exceeds this threshold. To conclude, we do not link each and every particle. Usually there are more trajectory particles than references and obstacles in the scene. Therefore the complexity reduces to a maximum of $O(2n + rn + on) \approx O(n)$ if we link all n particles only to their successor particle, to all o obstacles, and to all r references.

D. Numerical Integration

The commonly used Euler integration in simulations suffers from numerical instabilities and drift, especially with bigger step sizes. We use a Runge-Kutta method instead to overcome the issues in numerical integration. We estimate the position $p(t+1)$ as

$$p(t+1) = p(t) + \frac{1}{6} \cdot dt \cdot (v_{k1} + 2v_{k2} + 2v_{k3} + v_{k4})$$

with

$$\begin{array}{cc} p_{k1} = p(t) & v_{k1} = v(t) \\ p_{k2} = p(t) + \frac{1}{2} m v_{k1} & v_{k2} = v(t) + \frac{1}{2} m F_{k1} \\ p_{k3} = p(t) + \frac{1}{2} m v_{k2} & v_{k3} = v(t) + \frac{1}{2} m F_{k2} \\ p_{k4} = p(t) + \frac{1}{2} m v_{k3} & v_{k4} = v(t) + \frac{1}{2} m F_{k3} \end{array}$$

The orientation is calculated analogously from the torques and the angular velocities.

As soon as the positions and orientations of the reference particles are updated according to the new situation information, the trajectory starts to adapt to it. The adaption ends if the kinetic energy E in the system reaches a manually set threshold of $E < \epsilon$. The kinetic energy E of the system is calculated from the velocity and the angular velocity of all particles as

$$E = \sum_i \frac{1}{2} m v^2 + \sum_i \frac{1}{2} I \omega^2$$

IV. Experimental Results

A. Experiments

We tested our approach in an experimental setup using a Kuka LWR IV Robot equipped with a Schunk PG70 gripper. An overhead RGB-D camera (Softkinetic DepthSense 325) was used to detect objects in the workspace, which were identified through augmented reality (AR) markers. The trajectory was obtained while the user guided the robot through the task. A filtering was applied to the trajectory to remove neighboring samples with zero distance.

We evaluated our approach in various experiments. We performed a quantitative cross validation against human demonstrations in Experiment 1 similar to those in [17]. We also performed a qualitative validation of the approach in Experiment 2. A preliminary video, which shows the online adaption of a trajectory, is available on our website[1].

1) Experiment 1: The first experiment aims at a quantitative evaluation of the approach. Like [17], we went for a general grasping, which is often used in robotic applications. We demonstrated the task, denoted as T1, in 5 different ways (series S1 to S5) with 20 repetitions each. This results in $100 \cdot 100 = 10.000$ cross validations. The task T1 consists of the following five series:

- Series S1 is a straight and direct grasping movement.
- Series S2 is based on S1, but the whole scenario is rotated by 90° around vertical up vector.
- Series S3 is based on S1, but only the target is rotated by 90° around up.
- Series S4 is a grasping with arbitrarily chosen 3D orientations and positions of the robot and the target.
- Series S5 is is based on S1, but with an obstacle on the way to the target.

We did a pair-wise comparison of each demonstration and every possible reproduction. Therefore we took the references R^D_{input} of a demonstration (input) and adapted another demonstration to this new positions (output). This was achieved by setting the references R^D_{input} from the input as references for reproduction to the output $R_{\text{output}} = R^D_{\text{input}}$. Afterwards, we compared the reproduced output trajectory with the human demonstrated input trajectory and calculated two performance metrics. In analogy to [17] we calculated the mean squared difference MSD as

$$MSD = \frac{1}{N} \sum_{i=1}^{N} ||p_i^r - p_i||$$

where p_i^r is the position of the reproduced sample (output) and p_i is the input sample position. The calculated values give insight on how close the reproduced trajectory is to a human demonstration.

We also calculated the correlation coefficient R^2 as

$$R^2 = \frac{\sum_{i=1}^{N} (p_i - \bar{p}) \cdot (p_i^r - \bar{p}^r)}{\sqrt{(\sum_{i=1}^{N} (p_i - \bar{p})^2) \cdot (\sum_{i=1}^{N} (p_i^r - \bar{p}^r)^2)}}$$

where $\bar{p}$ and $\bar{p}^r$ are the arithmetic means of p_i and p_i^r. The correlation coefficient indicates how similar a reproduced path is to a human demonstration.

Both metrics require that the input vectors being the same length. To guarantee that, we interpolated the shorter trajectory using a Hermite spline.

2) Experiment 2: The second experiment was meant to evaluate the qualitative results of the approach. Therefore we provided various demonstrations, which were adapted to new situations afterwards. The demonstrations were divided into simple (T2) and a more complex task (T3). The goal of the experiments is the reproduction of the tasks with changed orientations and positions of the targets while the characteristic movements are kept. In T2 we demonstrated a bow to a target and an arrowhead-like movement at the target

[1]http://ai3.uni-bayreuth.de/projects/introp

(Figure 4). In the adaption phase, we displaced and rotated the targets. In T3 we demonstrated the ironing of a piece of cloth to the robot. Afterwards we displaced and rotated the iron and the ironing board (Figure 6).
Since robots often work in places with a highly dynamic environment, we also investigated the capabilities to adapt a demonstration to a new situation in the presence of obstacles in task T4. Therefore we placed an obstacle in the way of an adapted trajectory (Figure 5).

B. Results

Regarding Experiment 1, for the case of the 100 self-mappings, where a demonstration is adapted to identical references, we consequently receive $MSD = 0$ and $R^2 = 1$. The system will act like in playback mode, which was one of our requirements in Section II since it perfectly reflects the users intended trajectory.
Tables I and II show the arithmetic mean of the performance metrics for each group of input / output series. In

TABLE I
MSD FOR ADAPTED SERIES OF T1

	input				
output	S1	S2	S3	S4	S5
S1	0.000921	0.000532	0.00293	0.00316	0.0118
S2	0.00064	0.000212	0.00475	0.00322	0.0141
S3	0.00371	0.00495	0.00155	0.00592	0.00784
S4	0.00444	0.00127	0.00655	0.00109	0.016
S5	0.0123	0.00663	0.00555	0.0143	0.00116

TABLE II
R^2 FOR ADAPTED SERIES OF T1

	input				
output	S1	S2	S3	S4	S5
S1	0.995	0.988	0.938	0.968	0.926
S2	0.994	0.998	0.911	0.962	0.905
S3	0.955	0.808	0.974	0.918	0.961
S4	0.96	0.967	0.901	0.989	0.892
S5	0.928	0.817	0.956	0.887	0.987

both tables, we can recognize reasonable results of our approach. We receive pretty low MSD, which indicates only few aberration to the human demonstrations. But more important, we receive a minimum correlation coefficient of $R^2 = 0.808$ and a maximum of $R^2 = 0.994$ off the diagonal. Furthermore 84% show a value of above 0.9 This means, the overall shape of the reproduction shows great similarity to the human demonstrations. Of course, the coefficients are lower on S4 and S5. The task is more complicated and therefore, the human provided demonstrations differ more from each other.

We can recognize that the algorithm is able to adapt a demonstration to a new situation with changed starting position of the robot and with arbitrary changed object positions and orientations even in the presence of an obstacle.

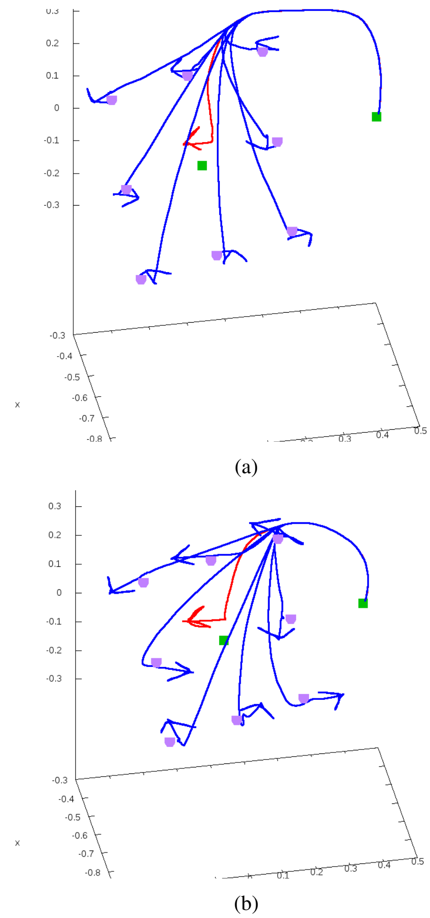
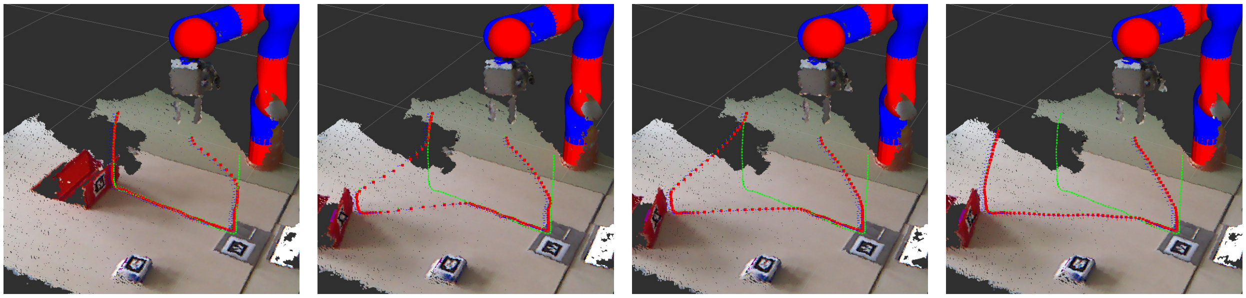
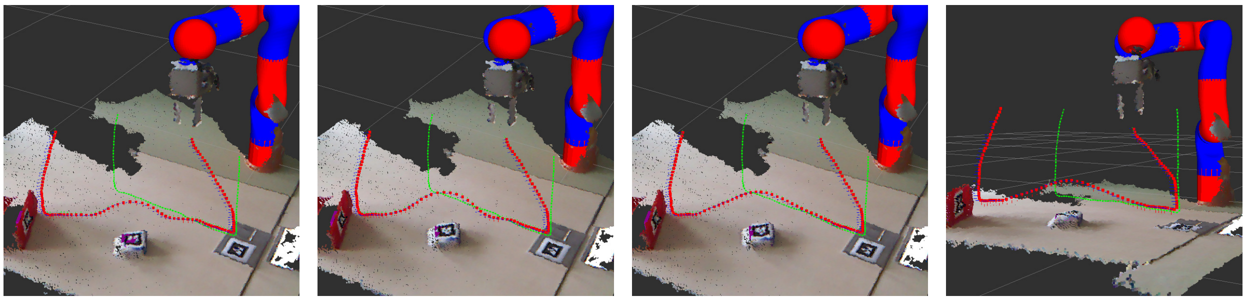


Fig. 4. 3D Demonstration (trajectory red, objects green) and reproductions with changed positions and orientations (trajectory and objects blue) of T2 with $d_{max} = 0.2$ (a) and $d_{max} = 0.4$ (b)

The quality of the data shows, that the reproduction is similar to the motion a human would perform under these conditions.
Figures 4 to 6 show the qualitative results of the approach. As one can see in Figure 4, the typical bow-motion of T2 is maintained, as well as the arrowhead, which was drawn at the target location. While the arrow adapts to the orientation of the target marker, the bow is stretched. If we alter the the parameter d_{max}, the transfer-bow will slightly change and lose its straight motion with increasing d_{max}, but the adaption will speed up. If we lower d_{max}, the adaption will take longer since less particles are directly linked to the environment, but as long as some of the particles are linked, the target motion is maintained. The figures additionally show the adaption to different z-levels, which is also handled very well by the algorithm.
In Figure 6 one can see the adaption of the ironing task T3. For better visualization, the robot model is reprojected into the scene. Part (a) shows the demonstration process, while parts (b) - (d) show the reproduction of the adapted trajectory. Parts (e) - (h) show the step-wise adaption of the trajectory. One can recognize that the particles, which are directly linked to the environment, are adapted at first. By

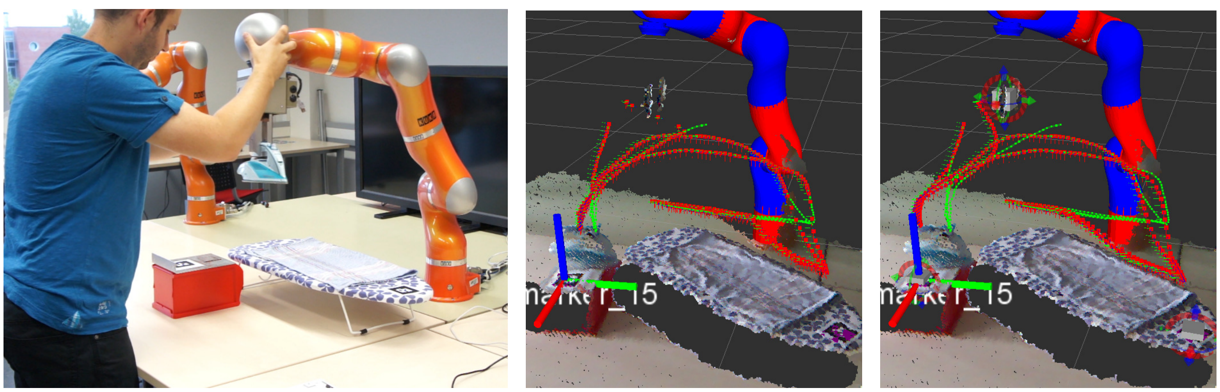


(a) (b) (c) (d)

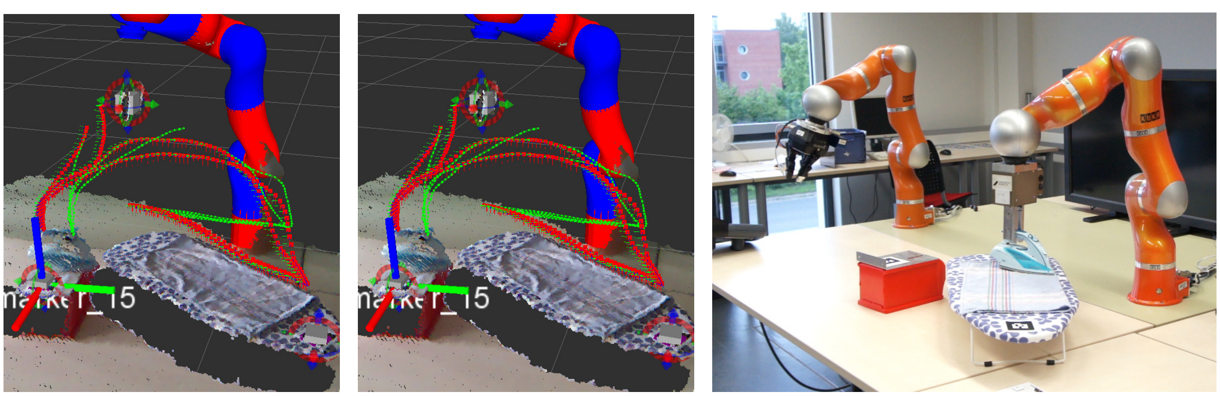


(e) (f) (g) (h)

Fig. 5. T4: The first row shows the evolving of a demonstration (green trajectory) to a new situation (red trajectory). The second row (e - h) shows the evolution of the trajectory in the presence of an obstacle.



(a) Demonstration of the task (b) (c)



(d) (e) (f) Reproduction of the task

Fig. 6. The task is demonstrated to the robot via kinesthetic teaching (a), adapted to the new situation (b-e) and reproduced by the robot (f)

and by the remaining particles adapt to the new situation. It is noteworthy, that d_{max} is limited to $d_{max} = 0.35m$, since the ironing board is that big. The amount of displacement is limited due to the workspace of the robot. Since the tool operations are attached to the particles, the robot is able to manage pick and place operations, like with the iron.

In Figure 5 we can see the adaption of the particles in the case of an obstacle (T4). The trajectory is adapted to the new situation and immediately bend around the obstacle to avoid a collision. By and by the bulge of the trajectory is smoothed out, resulting in a cleaner motion. In our case, we only consider the robot tool center point, but the approach could easily be extended to take into account a complete robot model.

The figures of the qualitative reproduction confirm the quantitative results. The produced trajectories have obviously high similarity with the human demonstrated, while simultaneously being able to solve the task.

Although we applied the mentioned filtering of the trajectory, we can still observe a positive effect of the time-discrete sampling. Motions, that need precision, are demonstrated more slowly, yielding in more samples with lower distance in between. Motions that are executed faster and more vague, like the bowed transfer in T2, have weaker links. If the user demonstrates parts very fast, he is most likely aware of being inaccurate. Since these are the parts, which are not required to be reproduced very precise, the system should be allowed to adapt these parts stronger. And this is exactly how the system behaves. The slower parts have stronger links, hold together more tightly and are therefore reproduced more exactly than the loosely linked particles due to fast user motions.

V. Conclusions and Future Work

This paper presents a new approach to programming by demonstration through a single demonstration. The approach regards every sample of a trajectory and every reference as an oriented particle in a particle simulation. We defined the necessary forces and torques to adapt a demonstration to a new situation. The approach adapts a demonstration based on the defined references. In contrast to existing work, the algorithm only expects a single demonstration and is capable of adapting the orientation of the robot. No manual post-processing is required and a collision avoidance is integrated. The adaption works online, i.e. the trajectory is updated whenever the references change position or orientation. We provided qualitatively and quantitatively results by comparing the adaption with human ground truth data.

The experiments show, that the approach is able to learn and generalize a task from a single demonstration. Although the tasks may seem trivial in robotics, one has to remember the constraints of the algorithm. The system only expects one demonstration from the user and it is not aware of what has been shown. It does not recognize, that for example a pick and place task has been demonstrated. It is also completely unaware of the effects it is triggering. There is neither deduction of the actions nor a priori knowledge. Nevertheless, the system is able to adapt a demonstration to a new situation with just one demonstration and a set of references.

The approach can be easily extended by additional forces and torques to match the requirements of other robots and their applications. I.e. a collision avoidance force could be added which takes the complete robot model into account. Further work may focus on the initialization of the particle links and on the integration of the approach into our behavior-based system [23] [21] [22].

References

[1] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469–483, May 2009.

[2] D. H. Grollman and O. C. Jenkins, "Incremental learning of subtasks from unsegmented demonstration," *2010 IEEE/RSJ Intelligent Robots and Systems (IROS)*, pp. 261–266, Oct. 2010.

[3] S. Calinon and Z. Li, "Statistical dynamical systems for skills acquisition in humanoids," *Humanoids*, 2012.

[4] T. Cederborg, A. Baranes, and P.-Y. Oudeyer, "Incremental local online Gaussian Mixture Regression for imitation learning of multiple tasks," *Intelligent Robots and Systems (IROS)*, Oct. 2010.

[5] A. Wrede, Sebastian Emmerich, Christian Grünberg, Ricarda Nordmann, A. Swadzba, and J. Steil, "A User Study on Kinesthetic Teaching of Redundant Robots in Task and Configuration Space," *Journal of Human-Robot Interaction*, vol. 2, no. 1, pp. 56–81, 2013.

[6] L. Rozo, S. Calinon, D. Caldwell, P. Jim, C. Torras, and I. D. Rob, "Learning Collaborative Impedance-based Robot Behaviors," *AAAI Conference on Artificial Intelligence.*, 2013.

[7] G. Ye and R. Alterovitz, "Demonstration-guided motion planning," *International Symposium on Robotics*, 2011.

[8] S. Schaal, J. Peters, J. Nakanishi, and A. Ijspeert, "Learning movement primitives," *Robotics Research*, pp. 1–10, 2005.

[9] J. Peters and S. Schaal, "Policy learning for motor skills," *Neural Information Processing*, pp. 233–242, 2008.

[10] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal, "Learning and generalization of motor skills by learning from demonstration," *Robotics and Automation (ICRA)*, pp. 763–768, May 2009.

[11] R. Dillmann, O. Rogalla, M. Ehrenmann, R. Zollner, and M. Bordegoni, "Learning from Human Demonstration and Advice: the machine learning paradigm," *Robotics Research*, vol. 9, pp. 229—-238, 2000.

[12] R. Zollner and M. Pardowitz, "Towards cognitive robots: Building hierarchical task representations of manipulations from human demonstration," *Robotics and Automation (ICRA)*, 2005.

[13] R. Zollner, T. Asfour, and R. Dillmann, "Programming by demonstration: dual-arm manipulation tasks for humanoid robots," *Intelligent Robots and Systems (IROS)*, vol. 1, pp. 479–484, 2004.

[14] S. Niekum, S. Osentoski, G. Konidaris, and A. G. Barto, "Learning and generalization of complex tasks from unstructured demonstrations," *Intelligent Robots and Systems (IROS)*, pp. 5239–5246, Oct. 2012.

[15] P. E. Rybski, K. Yoon, J. Stolarz, and M. M. Veloso, "Interactive Robot Task Training through Dialog and Demonstration," *Forbes*, 2007.

[16] H. Mayer, I. Nagy, and A. Knoll, "Adaptive control for human-robot skilltransfer: Trajectory planning based on fluid dynamics," *Robotics and Automation (ICRA)*, pp. 10–14, 2007.

[17] Y. Wu and Y. Demiris, "Towards One Shot Learning by imitation for humanoid robots," in *Robotics and Automation (ICRA)*, 2010, pp. 2889–2894.

[18] J. Schulman, J. Ho, C. Lee, and P. Abbeel, "Learning from Demonstrations Through the Use of Non-Rigid Registration," *International Symposium on Robotics Research*, 2013.

[19] H. Friedrich, R. Dillmann, and O. Rogalla, "Interactive robot programming based on human demonstration and advice," *Sensor Based Intelligent Robots*, 1999.

[20] M. Müller and N. Chentanez, "Solid simulation with oriented particles," *ACM SIGGRAPH 2011 papers on - SIGGRAPH '11*, vol. 1, no. 212, p. 1, 2011.

[21] C. Groth and D. Henrich, "Multi-Tasking of Competing Behaviors on a Robot Manipulator," in *Intelligent Robots and Systems (IROS)*, 2013.

[22] C. Groth and D. Henrich, "Single-Shot Learning and Scheduled Execution of Behaviors for a Robotic Manipulator," in *41st International Symposium on Robotics*, 2014.

[23] C. Groth and D. Henrich, "One-shot Robot Programming by Demonstration by Adapting Motion Segments," in *Robotics and Biomimetics*, 2014.