International Journal of Social Robotics manuscript No.
(will be inserted by the editor)

Keyframe-based Learning from Demonstration

Method and Evaluation

Baris Akgun · Maya Cakmak · Karl Jiang · Andrea L. Thomaz

Received: November 08, 2011 / Accepted: date

Abstract We present a framework for learning skills from novel types of demonstrations that have been shown to be desirable from a human-robot interaction perspective. Our approach –Keyframe-based Learning from Demonstration (KLfD)– takes demonstrations that consist of keyframes; a sparse set of points in the state space that produces the intended skill when visited in sequence. The conventional type of trajectory demonstrations or a hybrid of the two are also handled by KLfD through a conversion to keyframes. Our method produces a skill model that consists of an ordered set of keyframe clusters, which we call Sequential Pose Distributions (SPD). The skill is reproduced by splining between clusters. We present results from two domains: mouse gestures in 2D and scooping, pouring and placing skills on a humanoid robot. KLfD has performance similar to existing LfD techniques when applied to conventional trajectory demonstrations. Additionally, we demonstrate that KLfD may be preferable when demonstration type is suited for the skill.

Keywords Learning from Demonstration · Kinesthetic Teaching · Human-Robot Interaction · Humanoid Robotics

1 Introduction

The goal of Learning from Demonstration (LfD) is to enable humans to teach a robot new skills by showing successful examples [5]. There are various ways to provide these demonstrations. In this work we focus on *kinesthetic teaching*, in which a human teacher physically guides the robot in performing the skill, as shown in Fig. 1.

B. Akgun, M. Cakmak, K. Jiang, A.L. Thomaz
School of Interactive Computing, Georgia Institute of Technology
Atlanta, Georgia 30332
Tel.: +1-404-894-8591
Fax: +1-404-894-2223
E-mail: {bakgun3,maya,k.jiang,athomaz}@cc.gatech.edu

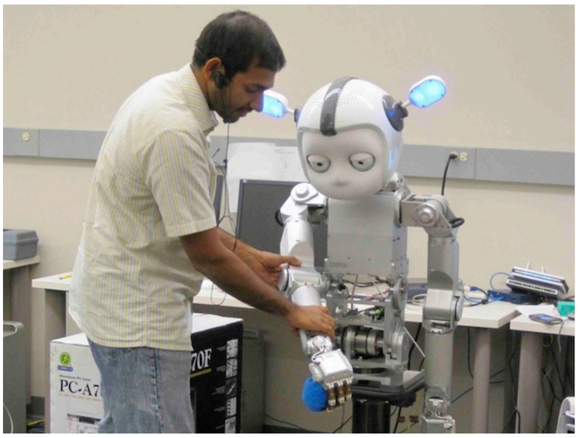


Fig. 1 Simon interacting with a teacher, learning a new skill via KLfD.

Kinesthetic teaching has several advantages. Since the teacher directly manipulates the robot there is no correspondence problem and demonstrations are restricted to the robot's kinematic limits (*e.g.* workspace, joint limits). Moreover, extra instrumentation (motion capture or teleoperation devices) is not necessary. However, kinesthetic teaching could pose challenges for everyday users who do not have experience manipulating a robot with many degrees of freedom.

In the typical kinesthetic teaching interaction, and in most LfD interactions, each demonstration is an entire state *trajectory*, that is, the teacher provides a continuous uninterrupted demonstration of the skill to the robot.

The assumption is that kinesthetic teaching will be particularly intuitive for end-users, but there have been relatively few user studies of existing methods. In previous work [2, 3], we analyzed LfD from an interaction perspective. We introduced an alternative to trajectory demonstrations, providing a sparse set of consecutive *poses* or *keyframes*, that achieve the skill when connected together. In multiple studies, we have concluded that both keyframe and trajectory input modes have their advantages, depending on the nature of the skill being taught. Moreover, for complex skills it is likely that portions of the skill are best taught with keyframes and other portions with trajectories. Such skills can be taught through what we call *hybrid* demonstrations.

Our goal is to improve LfD from a Human-Robot Interaction (HRI) perspective, by enabling end-users to use different the demonstration types at different points in time. Our intuition is that people will have differing preferences, but it may also be the case that some tasks/skills lend themselves to a particular type. In this paper we present a system that can learn in a unified way regardless of demonstration type (trajectory, keyframe or hybrid). We show how to process data collected with different input methods. Our approach converts all demonstrations into keyframes and produces a skill model that is Sequential Pose Distributions (SPD). We refer to the entire learning approach as Keyframe-based LfD (KLfD).

We first present some background that motivates our approach and provide details on the different steps of the KLfD framework. Next we present two evaluation domains and describe data collection. In Sec. 6 we give example outcomes of the method for different domains and input types. We present a quantitative comparison of KLfD with a baseline LfD technique, and a comparison of different input types. We find that KLfD is on par with a conventional LfD technique when using traditional trajectory demonstration input, and that KLfD performs best when the input type is suited for the particular skill. Finally we discuss possible extensions to the framework, suggesting ways to replace modules of our KLfD pipeline with other techniques.

2 Background

Traditional LfD techniques work with demonstrations that are continuous sequences of points in the state space, referred to as *trajectories*. Typically start and end points of a demonstration are explicitly demarcated by the teacher, and the robot records (with a sufficiently high frequency) the change of the state between these two events.

Demonstrations often consist of arm joint and/or end-effector trajectories [9, 16]. Some also consider the configuration of the end-effector with respect to the target object of the skill [6, 14]. Most studies subsample the recorded data with a fixed rate [4, 6]. Demonstrations are often time warped such that a frame-by-frame correspondence can be established between multiple demonstrations [16].

2.1 Learning from Demonstration Techniques

There are several methods for learning skills; most can be categorized as either direct policy learning or cost/reward learning. Dynamical system approaches such as Stable Estimator of Dynamical Systems (SEDS) [18] and Dynamic Movement Primitives (DMP) [24] as well as mixture models (*e.g.* Gaussian Mixture Models as in [10]) are policy learning methods. On the other hand, inverse reinforcement learning (IRL) [1] or apprenticeship learning and inverse optimal control (IOC) [25] are cost/reward learning methods. The policy is derived from the cost/reward afterwards.

These methods were designed for different purposes and each have their pros and cons. Apart from GMMs and DMPs, most require many training samples which is not suitable for short-duration HRI settings. DMPs and GMMs have either implicit or explicit time dependency. Most methods either cannot handle cyclic skills or need reformulation to do so.

2.2 Keyframes

There have been previous uses of keyframe-related ideas in other fields. Keyframes have been used extensively in the computer animation literature [23]. The animator creates important frames in a scene and the software fills in between. In the LfD setting, an earlier work [21] utilizes via-points, which are similar to keyframes. These are extracted from continuous teacher demonstrations with the method proposed in [27] and updated to achieve the demonstrated skill. A recent approach is to only record keyframes and use them to learn a constraint manifold for the state space in a reinforcement learning setting [7]. Whole body grasps for a simulated humanoid are learned in [17] by forming template grasp demonstrations via *keyframes*, which are the start/end points of a demonstration and the points of contact and points of lost contact with the objects.

2.3 An HRI perspective on LfD

LfD is motivated by the goal of robots learning from end-users. A survey of LfD work [5] shows a vast range of different input schemes that lead to very different interactions for the end-user: teleoperating a robot, performing a task in a motion capture setting, performing the task uninstrumented, moving a robot kinesthetically to provide learning data. By and large, the field lacks an understanding of the usability of the assumed input mechanisms of various LfD techniques. This is our overall research agenda, to create LfD techniques that end-users find natural and intuitive.

In this work we focus on one popular input mode for LfD–kinesthetic teaching. Human-robot interaction has not been a focus of prior work on kinesthetic teaching, but there are a few examples. In [28], kinesthetic teaching is embedded within a dialog system that lets the user start/end demonstrations and trigger reproductions of the learned skill with speech. A modification to the kinesthetic teaching interface is kinesthetic correction [8], where the teacher corrects aspects of a learned skill in an incremental interaction by using a subset of joints in subsequent demonstrations.

3 Keyframe and Hybrid Demonstrations

In previous work, we proposed learning from demonstration inputs that are much more sparse than traditional trajectories [3]. We refer to these as *keyframe demonstrations* (KD). A KD is a sequence of critical points in the state space (*i.e.* poses) such that visiting each keyframe allows the robot to achieve the skill that is being taught. We refer to conventional demonstrations which are continuous sequences of points in the state space as *trajectory demonstrations* (TD).

With a series of user studies, we evaluated keyframe demonstrations against trajectory demonstrations from an HRI perspective, revealing a set of advantages and disadvantages for each [3]. Trajectory demonstrations were more intuitive for naive users, and allowed teaching complex skills where speed information is important. However, it was hard for users to move a high dimensional robot arm smoothly, requiring more practice and often resulting in noisy and undesirable movements. Keyframe demonstrations, on the other hand, were not affected by unintended, noisy motions. In addition, they provide a compact representation that allows generalization and flexibility and facilitates temporal alignment. A drawback of keyframe demonstrations is the lack of timing and speed information for keyframe poses.

In either LfD interaction the teacher needs to indicate the start and end of demonstration through some modality. We choose to use speech as it is hands-free. For a trajectory demonstration the teacher indicates the start, moves the robot, and then indicates the end. For keyframe demonstrations the teacher uses additional speech commands, while moving the robot, to indicate the keyframes.

Keyframe demonstrations allow the teacher to freely manipulate the robot and carefully configure it before recording the keyframes of the demonstration. Unlike trajectory demonstrations, this allows collecting demonstrations free of movement noise and mistakes. On the other hand, demonstrating complex curved movements requires a large number of keyframes when using keyframe demonstrations. We have proposed that *hybrid demonstrations* (HD), which can have both trajectory or keyframe segments, combine the advantages of both types of demonstrations.

In this article we have two experimental domains, detailed in Sec. 5, one of which is a simple 2D letter drawing domain. This domain is used for visualization throughout the description of our implementation in Sec. 4. The three demonstration types are illustrated in Fig. 2 with example demonstrations of the capital letter P, represented as a sequence of points in 2D. A trajectory demonstration involves completing the letter in a single motion. A keyframe demonstration highlights important way points of the motion. And one possible hybrid demonstration communicates the straight-line motion with two keyframes, and then the more complex curve with a trajectory.

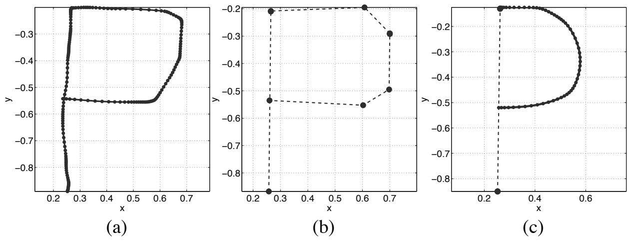


Fig. 2 Sample demonstrations of the letter P in 2D. (a) Trajectory demonstration (TD) (b) Keyframe demonstration (KD) (c) Hybrid demonstration (HD).

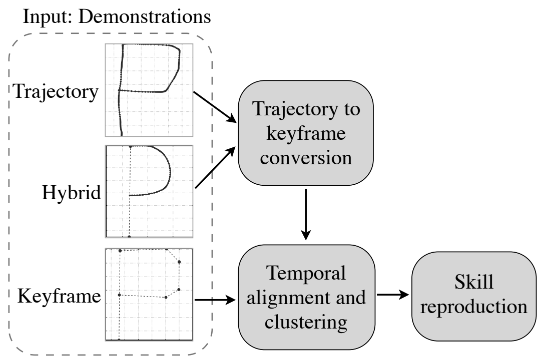


Fig. 4 Overview of the steps involved in KLfD.

4 Keyframe-based Learning from Demonstrations

Traditional LfD techniques are designed for continuous trajectories and cannot be directly applied to keyframe or hybrid demonstrations (*i.e.* sparse trajectories). Thus, our aim in this paper is to introduce a framework that can handle such varied types of input. In this section we present the KLfD implementation details. An overview of the steps involved in KLfD is given in Fig. 4. Note that individual steps can be implemented differently and we will briefly provide a few alternatives at the end of each step. For illustrative purposes we use 2D data for the capital letter P throughout this section. Details on how the data are generated is given later in Sec. 5.1.

4.1 Trajectory to Keyframe Conversion

Our framework supports input of trajectory, keyframe, or hybrid demonstrations. For trajectory and hybrid demonstrations, we add a preprocessing step to convert trajectories into keyframe sequences. To do so, we use the Forward-Inverse Relaxation Model (FIRM) [27]. Starting with the end-points

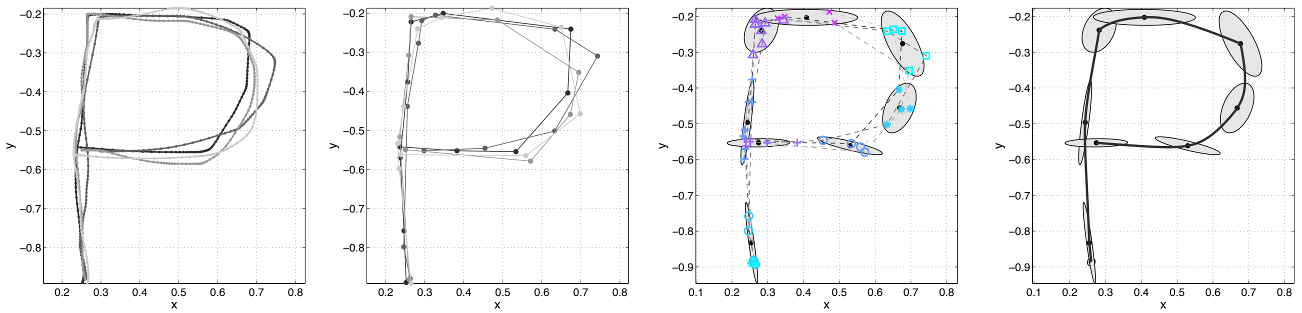


Fig. 3 Illustration of the steps in learning with keyframes. (a) Four demonstrations of the letter P given as continuous trajectories in 2D (b) Data converted to keyframes (c) Clustering of keyframes and the reulting SPD (d) Trajectory produced from the learned model.

of the trajectory as keyframes, we generate a new trajectory based on a fifth order spline on the keyframe positions and velocity and acceleration at that keyframe position. This is in essence leveraging the skill reproduction method described later in Sec. 4.3, which can be seen as an extension of Lowe's method [20].

If velocity and acceleration data is unavailable from the demonstration itself, we take the smoothed first and second derivatives of the demonstrated trajectory respectively (using a Gaussian filter). [1] We then compare the original trajectory and generated trajectory to locate the point which has the largest Euclidean discrepancy at any given time. This point is added as another keyframe and the process is iterated until the generated and target trajectories are within an error threshold. The threshold is determined empirically and can be domain specific. Fig. 3(b) shows keyframes obtained from the four trajectory demonstrations of the letter P shown in Fig. 3(a).

4.1.1 Alternative Approaches

There are other methods for extracting keyframes in video processing (*e.g.* [19]), motion capture and graphics (*e.g.* [15]) literatures. The advantage of our choice is that it uses the existing skill reproduction mechanism. This forces the reproduced skill to be closer to the demonstrations since the difference between the model trajectory and demonstrations is minimized. Other methods try to find salient points in a trajectory, which may or may not be relevant during reproduction of the skill.

4.2 Aligning and Clustering

The purpose of this step is to come up with a skill model given multiple keyframe sequences. We call this model Sequential Pose Distributions (SPD).

[1] Using velocity and acceleration data along with position data helps to keep some of the dynamics of the demonstration.

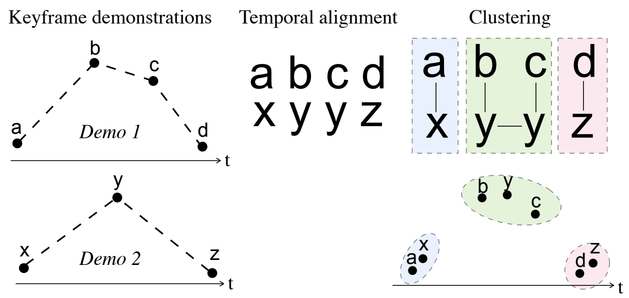


Fig. 5 Illustration of the alignment and clustering process.

Given several demonstrations of a skill, one common problem for LfD techniques is to temporally align the demonstrations before using them to build a model.[2] Dynamic Time Warping (DTW) is a widely used method for aligning two sequences at a time, whereas we need a general and order-independent method for aligning multiple keyframe demonstrations.

For this, we keep an average alignment to which we align all the sequences in an iterative process and keep an alignment pool (a set of previously aligned sequences). The average and the alignment pool are initialized with the lowest cost pair. After that, the next sequence is selected based on the lowest pairwise DTW cost between the aligned and not aligned sequences. The average and the pool are then updated with this sequence. Iterations are repeated until all the sequences are aligned.

After aligning, we cluster together any keyframes that are aligned to the same keyframe from another demonstration. This can be considered finding connected components in a graph which connects all keyframes that are aligned together through DTW. An illustrative example is given in Fig. 5. The outcome of this step is the learned SPD, which corresponds to the keyframe means and covariances of each cluster in sequence. Fig. 3(c) shows the clusters formed from the keyframe demonstrations in Fig. 3(b).

[2] Not all LfD techniques have this problem, *e.g.* [18].

4.2.1 Alternative Approaches

There are many alternatives for both aligning and clustering. The basic requirements are interaction time learning (no long training duration and scalability with increasing number of demonstrations) to allow interactive teaching and to come up with a skill model that can be used to generate motion. We present a few alternatives in this subsection.

The resulting connected components might be coarser than the user intended, especially if there is a spatial shift in the provided keyframes between demonstrations. We have not run into this problem in practice, but expect it to manifest itself with end-users. We can break these coarse connected components (represented as graphs with edge weights inversely proportional to the distance between keyframes) down by utilizing a threshold (*e.g.* a distance threshold) or using graph-cut methods. These solutions are at the expense of introducing additional parameters.

Multiple sequence alignment has been studied extensively in different domains such as biology (*e.g.* [11]) and speech recognition (*e.g.* [22]). Thus there are many methods that try to tackle the problem. Many methods follow an iterative approach that utilizes a dynamic programming approach between two-sequences (*e.g.* [22] or our described method) or Hidden Markov Models (HMM) (*e.g.* [12]). HMMs offer additional features such as inference (*e.g.* for recognition) which makes them attractive at the cost of additional computation time. However, our approach was preferred for its simplicity and the qualitatively satisfactory alignments for both keyframes and trajectories. Note that optimal multiple sequence alignment is an NP-Complete problem.

We do not need to explicitly align keyframes first to cluster them and in fact the alignment itself can be the result of clustering. A basic idea is to either choose the median (more generalizability) or maximum (more conservative) number of keyframes amongst the demonstrated sequences, use the demonstrations with the selected number to initialize the clusters and map the keyframes in other demonstrations to the clusters based on distance while respecting the ordering.

4.3 Skill Reproduction

Given the SPD, we use a fifth order spline to reproduce the learned skill. The spline is used to calculate states (*e.g.* positions) given time. This is motivated by the work in [13], which showed that human point-to-point motions resemble minimum-jerk trajectories and a fifth order spline is the function that minimizes the jerk cost.

We fit the spline through the means of the pose distributions.A fifth order spline has 6 unknowns. We use the start and end positions (obtained from the SPD), velocities and accelerations,to calculate these. For keyframe demonstrations, we assume zero initial and final velocities and accelerations, which yields a straight line in the state space. For trajectory demonstrations, we use the mean velocities and accelerations at the cluster centers, calculated from the input trajectories. The other component is the duration between two keyframe clusters. For keyframe demonstrations, we assume a constant average velocity to estimate this and for trajectory demonstrations, we use the average duration seen in the input trajectories. This splining will result in C^2 continuity at the keyframes and C^∞ elsewhere[3].

In Fig. 3(d) we show the trajectory reproduced with the described method from the model in Fig. 3(c). Note that there seems to be non-smooth transition on some of the keyframes on the generated letter *P*. This is due to the low velocity and acceleration of the demonstrations.

4.3.1 Alternative Approaches

There are other methods to calculate velocities and accelerations for spline points. A common approach used in graphics is to estimate them by utilizing previous and next keyframes. Another approach is to optimize the minimum jerk cost with respect to the velocities and accelerations given durations and positions, as done in [26]. Another approach is to learn a local model for velocities and accelerations instead of using the average.

We can also use other methods to move in between sequential poses (*i.e.* keyframes). We can pose it as another optimal control problem in which we relax the condition of passing through all the keyframes but keep the jerk cost and additionally penalize deviation from the distribution means by utilizing the distribution covariances.

With most of the methods, we are going to get a trajectory (*e.g.* as opposed to a complete policy) in the end. In practice, a controller is needed to follow this trajectory or stay close to it as possible while obeying other constraints. This is not a focus of our work at this point.

5 Data

In this section we describe the two domains used for evaluating our KLfD method.

5.1 Letters in 2D

Part of our evaluation is performed with 2D mouse gesture data, collected with a Java Applet (Fig. 6). The applet allows for collecting all three types of demonstrations (TD, KD, HD). A single click on the applet creates a keyframe, while dragging the mouse results in a trajectory segment.

[3] C^k continuity for a function means that the function's $1 \ldots k$ derivatives exist and are all continuous

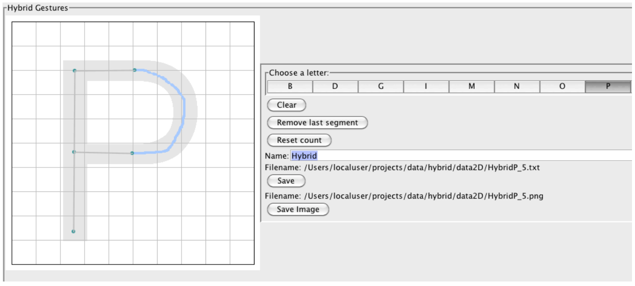


Fig. 6 Snapshot of the Java applet for collecting 2D mouse gesture data. The target letter to demonstrate is shown as a light grey template that is 38 pixels thick.

5.1.1 Skills

We evaluate six different skills corresponding to the letters: B, D, G, M, O, and P. The letters were chosen to have a variety of combinations of straight and curved segments. For each skill we created an image that consists of the template of the letter. The template is a light gray image of the letter with an average thickness of 38 pixels.

The goal of the skills in the 2D domain is to stay as close to the center, or skeleton, of the letter template. The ground truth skeleton is determined automatically as follows: The template is converted to a binary image and morphological *thinning* operation is applied to it. This creates a one pixel thick skeletal image (e.g., the red line in Fig. 10). Next, a starting position is chosen on the skeleton that roughly matches where the demonstrator begins their demonstrations.

We use a modified depth-first search algorithm to create the ground truth trajectory from the skeletal image. Pixels on the skeleton are added based on a depth-first search which explores neighboring skeleton pixels clockwise starting from the bottom-left one. Starting from the initial pixel, points are added to the trajectory when a pixel is first explored and when backtracking leads to a pixel. The search concludes when all the skeletal pixels have been explored.

Our success metric for generated trajectories is the DTW alignment cost between generated trajectory and the skeleton goal path, normalized by the length of the generated trajectory. Since the generated trajectories might have variable velocity and the goal trajectory has constant velocity, we resample the generated trajectory so that any two consecutive trajectory points are separated by the same distance. In our metric we set the distance between any two trajectory points to be one pixel.

The presented implementation of KLfD utilizes DTW to learn the SPD skill model (see Sec. 4.2). This seems to bias the proposed comparison metric at a first glance since DTW involved in both. However, KLfD is trying to align with respect to the demonstrations and align only keyframes (highly sparse with respect to the dense trajectory generated for comparison). Moreover, the trajectory demonstrations are aligned using DTW prior to being input to the GMM+GMR method. Thus comparing the skeleton as the baseline with the proposed metric will not be biased.

5.1.2 Data Collection

The data is collected through the applet shown in Fig. 6 on a MAC PC using a generic USB optical mouse. Four demonstrations were collected with each demonstration type (TD, KD, HD) for each letter. The hybrid demonstrations were chosen based on intuition: straight portions were shown as keyframes and curved portions were shown as trajectories (*e.g.* see Fig. 11). All demonstrations started at the same point for each letter, based on intuition on the starting position that would be optimal for drawing the letter in one continuous motion. This corresponds to the leftmost of the bottommost pixels, except in the case of G, which is drawn starting from the topmost endpoint. All demonstrations were provided by one of the authors.

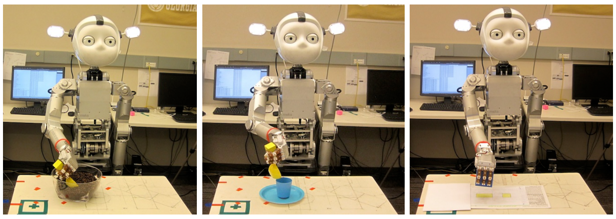
5.2 Robot Skills

In our second experimental domain, we evaluate our approach with table top manipulation skills on a humanoid robot. The robot platform used in this study is the Simon humanoid robot (Fig. 1). Simon, is an upper torso humanoid robot with two symmetric 7 degree of freedom (DoF) arms, 2 DoF torso and 13 DoF expressive head. The arms consist of series elastic actuators that allow safe interaction with humans and the environment. The right arm of the robot is kinesthetically guided during the demonstrations. The arms are gravity compensated to ease kinesthetic teaching.

5.2.1 Skills

We used the following three skills for evaluation:

- *Scooping:* In this skill, the robot holds an empty spoon and the teacher guides the arm to scoop as many coffee beans from a bowl as possible in one demonstration (Fig. 7(a)). This skill is demonstrated in trajectory mode. The success metric for scooping is the amount of coffee beans acquired (in grams).
- *Pouring:* In this skill, the robot holds a spoon full of coffee beans and the teacher guides the arm to pour as many beans from the spoon to a cup as possible in one demonstration (Fig. 7(b)). This skill is demonstrated in trajectory mode. The success metric for pouring is the amount of coffee beans successfully transferred into the cup (in grams). The initial content of the spoon is always the same.
- *Placement:* In this skill, the robot holds a block and the teacher guides the arm to place it to a designated area (Fig. 7(c)) with KD.



(a) Scoop (b) Pour (c) Place

Fig. 7 The three robot skills used in our evaluation.

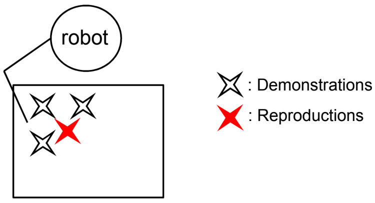


Fig. 8 Setup for data collection and evaluation on the robot.

5.2.2 Data Collection

The setup for collecting demonstrations is illustrated in Fig. 8. Each skill is demonstrated for three goal locations (the bowl location for scooping, the cup location for pouring and the designated target area for placement). Two demonstrations per location are recorded, resulting in a total of 6 demonstrations per skill. All demonstrations were provided by one of the authors. A different goal location is used for the evaluation of the reproduced skill.

The state recorded during demonstrations is time stamped coordinates and rotation quaternions of the end-effector with respect to the robot base frame, which is 8-dimensional. We pre-process the trajectory demonstrations before applying the methods described in Sec. 5. The demonstrations are converted to the object frame and filtered using a Gaussian filter with the cut-off frequency chosen as $2Hz$. The frame transformation is necessary such that there is correspondence between multiple demonstrations and filtering is necessary since the teacher demonstrations are inherently noisy. The frequency is chosen empirically based on the frequency amplitude spectrum of the data.

6 Results

In this section we provide qualitative and quantitative evaluations of the KLfD learning framework. We first provide example outcomes of learned SPD models on both domains. Next we provide a comparison of KLfD with an existing LfD technique on trajectory demonstrations. Finally we compare the outcomes of SPD models when used with three different types of input demonstrations (TD, KD and HD).

As a baseline for comparison, we chose the LfD method described in [10]. In this method, a Gaussian Mixture Model (GMM) is fit to the data using the Expectation-Maximization (EM) algorithm. Then, Gaussian Mixture Regression (GMR) is used for skill reproduction from the model. There are multiple reasons for this baseline choice. This method can be trained with a low-number of demonstrations and training can be done in interaction time. It can handle cyclic skills (assuming constant number of cycles) as well as point to point skills. Moreover, the GMR portion generates smooth trajectories. We use time-stamped positions as our states and query the GMR with a desired time vector to get a desired trajectory. We refer to this method as GMM+GMR. The trajectories are aligned in time using DTW prior to being input to this algorithm.

6.1 Sample outcomes

6.1.1 Letters

Fig. 9 shows the reproduced skill outcomes that result from KLfD and GMM+GMR for trajectory type input demonstrations on a subset of three letters in the 2D domain. Note that there seems to be a piece-wise linear effect. This is due to low velocity and acceleration and sharp turns inherent in the provided demonstrations (see Fig. 3(a)). Both approaches produce qualitatively similar outcomes given the same trajectory input data.

This baseline approach is not designed to handle sparse trajectories so we only provide KLfD output for keyframe and hybrid input data. Fig. 10 shows the outcomes of SPD skill models for keyframe type input demonstrations on all six letters in the 2D domain. Note that these models are obtained from multiple demonstrations (4). We see that the resulting trajectories resemble the intended letters. The piecewise linear appearance is due to our zero initial and final velocity assumption on keyframes. Comparing Fig. 10 and the top row of Fig. 9, we see that trajectory input results in learned models that look more similar to the intended letters, since the demonstrations themselves contain more information about the curved parts of the skills.

Fig. 11 shows the set of four hybrid demonstrations provided for the letter P and the SPD outcomes. Our KLfD method succeeds in learning an appropriate SPD model, despite the non-uniformity of the demonstrations. It can be argued that the resulting letter P is more similar to the intended one than any of the Ps in Fig. 9 or Fig. 10.

6.1.2 Robot skills

Fig. 12 shows the demonstrations provided for the scooping skill and the trajectories reproduced using GMM+GMR and SPD models. Two representative dimensions of the state-space are shown: the vertical dimension and the angle-component of the quaternion. The top row corresponds to pre-processed

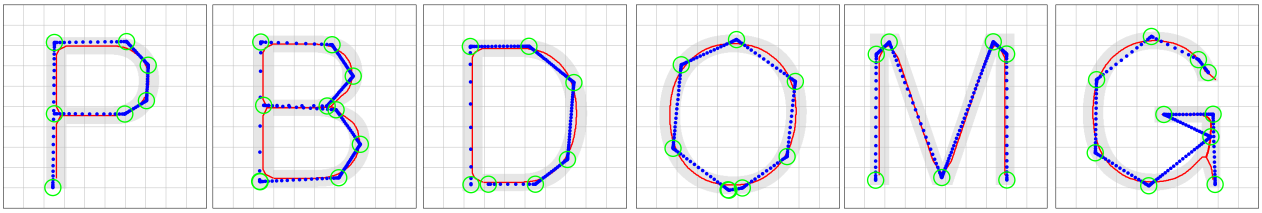


Fig. 10 Outcomes of KLfD with keyframe demonstration inputs for 6 skills in the 2D letter domain. The thin red line shows the skeleton of the letter that the teacher tries to demonstrate using the mouse. The thick lines show the reproduced trajectory.

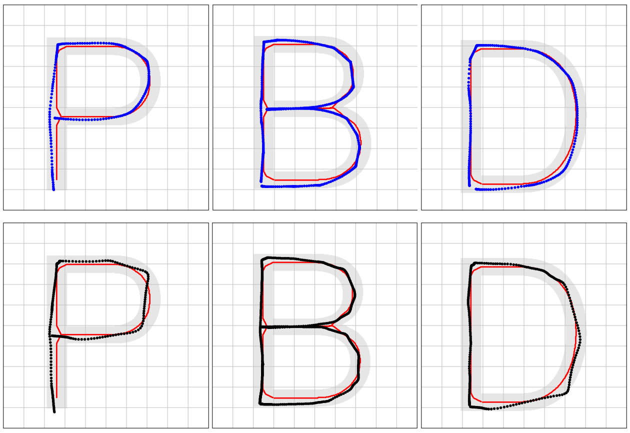


Fig. 9 Outcomes of KLfD (top row) and a baseline trajectory learning approach (GMM+GMR) (bottom row) with trajectory demonstration inputs for 3 skills in the 2D letter domain. The thin red line shows the skeleton of the letter that the teacher tries to demonstrate using the mouse. The thick lines show the reproduced trajectory.

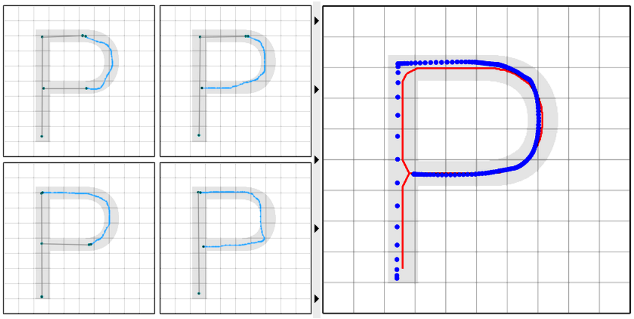


Fig. 11 Outcome of KLfD for hybrid demonstration inputs for the letter P. The red line shows the skeleton of the letter and the blue dots show the trajectory reproduced based on the learned skill.

teacher demonstrations and the extracted keyframes. Note that the data is highly varied and not aligned in time. The middle row shows the aligned trajectories (as described in section 4.2), the learned GMM and the resulting trajectory. The bottom row shows the aligned keyframes, the SPD model and the resulting trajectory. The algorithm for alignment is the same for trajectories and keyframes but the input data is different.

The vertical dimension (left column in Fig. 12) of the scoop captures the dip into the bowl. It can be argued that the variance is lower in the dipping portion. This is from the fact that all the demonstrations had this in common, *i.e.* this was the important part of the skill. Note that both of the methods generated similar resulting trajectories and captured the low variance of this portion.

A rotation quaternion represents a rotation around an axis. Specifically, the angle-component is the cosine of the half of the rotated angle. A single component of the quaternion by itself is not enough to capture all the rotation information of the end effector but gives a rough intuition. The resulting trajectories (right column in Fig. 12) show that there is nearly a monotonic change in this angle which is consistent with the scooping skill.

Fig. 13 shows the 2D projection of the keyframe demonstrations, the SPD and the generated trajectory for the placement skill. Note that the initial and final clusters have higher variance and variance lowers as the skill approaches the placement position. The algorithm identified 5 important regions for the skill. We can interpret these as the start and end of the skill, pre-placement position, safe retraction position and the placement position. During the placement demonstrations, the teacher was able to take his time to correctly align the block with the placement position, which was possible to due to the keyframe demonstrations.

6.2 Comparison with trajectory-based methods

Our framework accommodates trajectory demonstrations by converting them to keyframe demonstrations, as described in section 4.1. This can be viewed as a loss of information. In order to show that this loss does not effect the performance of the learned skill we first focus on trajectory demonstrations as the input, and quantitatively evaluate our framework in comparison to the baseline GMM+GMR.

6.2.1 Letters

A comparison of performance on the six letter skills is shown in Fig. 14(a). The success metric is alignment cost, as described in Sec. 5.1. For all six letters we find that SPD models produce letters that are closer to the template (*i.e.* have lower alignment cost with the template skeleton). In addition, we see that both methods produce skills that are more successful than the provided demonstrations of the skill.

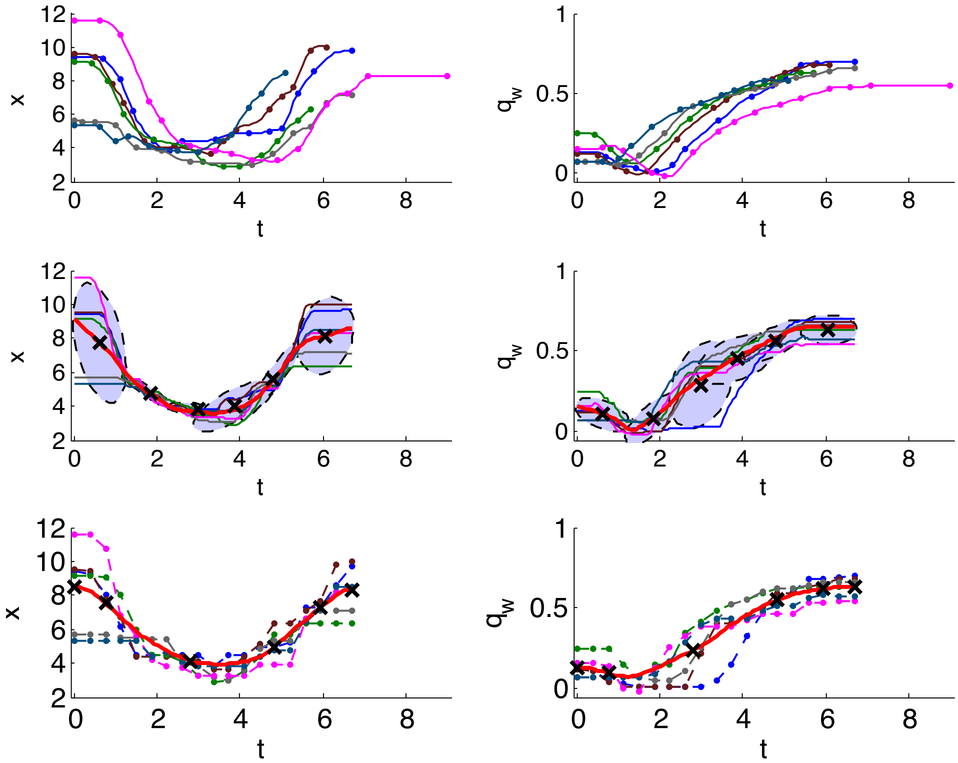


Fig. 12 The demonstrations and the learned trajectories for the x (verticals)) and the q_w (angle representation of the quaternion) dimensions of the scoop skill. Vertical axes correspond to the dimensions and horizontal axes correspond to time. Top row: Filtered and transformed (with respect to the object) raw trajectories and the extracted keyframes (dots). Middle Row: Aligned demonstrations and the learned trajectory (red) using GMM+GMR. The covariance between the dimensions and time is represented by the light blue ellipsoids and x-marks represent the centers of the GMMs. Bottom Row: Aligned keyframes (dots, dashed lines are to ease visualization) and the learned trajectory (red) using the keyframe method. The x-marks denote the means of the pose distributions.

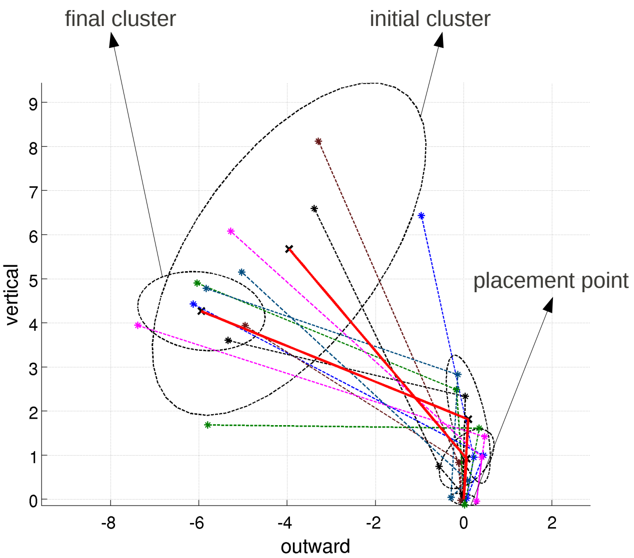
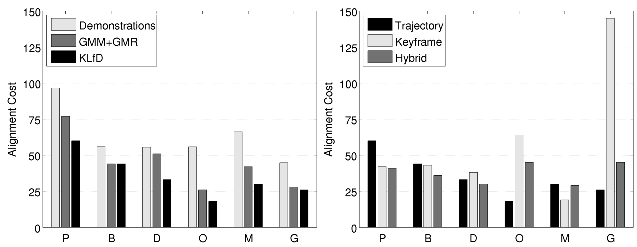


Fig. 13 The 2D projection of the placement demonstrations. The asterisks mark the demonstrated keyframes. The dashed-lines are given for visualization purposes. The ellipses represent the covariances and x marks represent the means of the pose distributions and the red solid line is the reproduced trajectory.



(a) (b)

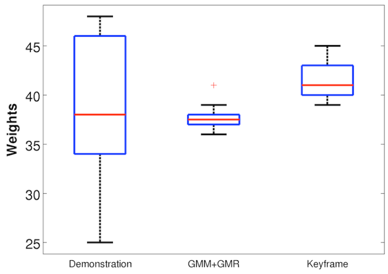
Fig. 14 Skill success in the 2D letter domain measured with costs for alignment with the template letter (lower cost means better alignment). (a) For skills learned with GMM+GMR versus with KLfD using trajectory type input demonstrations. (b) For skills learned with KLfD using three different input demonstration types. Note the KLfD bars in (a) are equivalent to Trajectory bars in (b).

6.2.2 Robot Skills

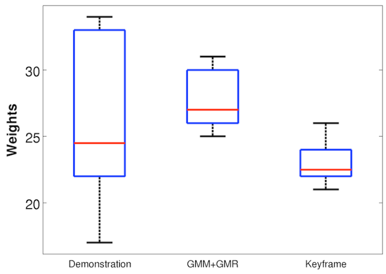
We compare KLfD and GMM+GMR on two robot skills: scooping and pouring. The success metric is bean weight, as described in Sec. 5.2. Demonstrated trajectories were played back on the robot three times per respective location (a total of 18 demonstrations) and the success metric is recorded for each. These results can be seen in Table 1 in the *Demonstra-*

Table 1 Comparison of the success of (i) provided demonstrations, (ii) trajectories learned with KLfD and (iii) with GMM+GMR on two skills. Values indicate weights in grams and standard deviations are given in parentheses. Note that demonstrations have 18 samples and learned models have 10.

	Scoop	Pour
Demonstrations	38.4 (7.3)	26.2 (5.8)
Learned skills (KLfD)	41.5 (2.0)	23.0 (1.7)
Learned skills (GMM+GMR)	37.8 (1.4)	27.8 (2.3)



(a) Weight of scooped coffee beans



(b) Weight of poured coffee beans

Fig. 15 Box-plots for the skill success measures comparing (i) replayed teacher demonstrations (18 samples), (ii) trajectory obtained with the model learned with the GMM+GMR method (10 samples) and (iii) with the KLfD method (10 samples) for two skills.

tions row. This is a sanity check on the data, showing that robot has seen successful demonstrations, so we expect the learned models to perform similarly.

The skills are learned with both methods and the reproduced skill is performed at a different target location (the red cross in fig. Fig. 8). Each learned model is executed 10 times. The results are reported in Table 1 and the descriptive statistics of the results can be seen as box-plots in Fig. 15.

These results show that the performance of both learning methods have success similar to the demonstrations. Moreover, they are similar to each other. In scooping, KLfD resulted in a more successful SPD model and vice-versa for pouring. The methods were not tuned with respect to any skill and parameters were chosen to be generic. This shows that our method is on par with a state of the art technique at building models from trajectory demonstrations.

Note that we make no claim of providing significant improvements over the GMM+GMR method. However, we get comparable results with a method that allows for keyframe or hybrid input data in addition to handling trajectories.

6.3 Comparison of demonstration types

Next, we look at the impact of input type (TD, KD, HD) on skill success in the 2D letters domain. This comparison, in terms of the alignment costs with the template skeleton, is shown in Fig. 14(b). We observe that hybrid demonstrations result in the best performance for the letters P, B and D, followed by keyframe demonstrations. Hybrid demonstrations have an advantage over keyframe demonstrations due to the ability to provide the curved parts of the letters more accurately. Trajectory demonstrations have the highest costs in these skills. This is mainly due to the difficulty of drawing a straight line when giving trajectory demonstrations.

For the letter O we see that trajectory demonstrations result in the best performance. This is again intuitive since this letter is entirely curved. At the other end of the spectrum, the drawing of letter M consists of only straight movements. As a result, we find that a pure keyframe demonstration results in the best alignment. For the hybrid demonstrations of these two letters, we intentionally tried to balance the use of keyframe and trajectory segments, even though the usage of hybrid demonstrations for these letters is less intuitive. For the letter G we see that trajectory demonstrations perform best, since the letter is predominantly curved.

Overall we see that the best KLfD performance results are achieved when the demonstration type is suited for the skill. In the 2D letter domains this implies using trajectory demonstrations for O and G, keyframe demonstrations for M and hybrid demonstrations for P, B and D. This confirms our intuition about the utility of being able to provide a variety of demonstration types to handle a range of skills.

7 Discussion

Our previous work, [3], motivated the alternative input modes, keyframe and hybrid demonstrations, for kinesthetic teaching in LfD. In this article we develop a framework for learning from these alternative types of demonstrations. One of the main strengths of the framework is that it handles all three types of input demonstrations. This allows a human teacher to use the input mode that is most comfortable to them or that they see most suitable for a given skill. In addition, this allows them to change their input mode over time, *e.g.* show some trajectory demonstrations and some keyframe demonstrations for the same skill.

Hybrid demonstrations are particularly strong as they allow the demonstration to be adapted to the particular parts of a skill. Typically skills involve multiple components. For instance it is natural for scooping and pouring to be demonstrated together. Parts of the skill that requires a complex motion of the spoon to collect the beans or to pour them accurately into the cup are suited for trajectory demonstrations. Whereas, the parts before, after or in between these movements are more suited for keyframes. This is analogous to the 2D skills corresponding to the letters P, B, D we considered in Sec. 6.3. We found that KLfD produces the best results with hybrid demonstration inputs for these skills. The hybrid demonstrations allow for traditional trajectory demonstrations, so there is an added benefit with hybrid demonstrations instead of a trade off.

We also have anecdotal evidence from the AAAI 2011 Learning from Demonstration Challenge that end-users come up with their own styles of teaching and learn and adapt quickly to the skill.[4] We believe that hybrid demonstrations will be beneficial for end-users to program their robots. Validating this with a user study is the next step in our future work with KLfD and hybrid demonstrations.

Sequential Pose Distributions (SPD) skills are modeled from keyframes. Thus trajectory or hybrid demonstrations need to be converted to keyframe demonstrations. It is important to ensure that this conversion is not detrimental to the success of the learned skill. Our results showed KLfD can learn SPD models that have performance on par with existing methods.

We also note some of the current limitations of KLfD. As mentioned throughout Sec. 4, there are several parameters that need to be chosen empirically. In this article these were domain specific. We found that one set of parameters worked well with multiple skills for a given domain. Another concern is that the zero velocity and acceleration assumption for demonstrated keyframes might be too restrictive for certain skills. Moreover, the current version of the algorithm does not leverage the pose distribution covariance. We further pointed out other methods that can be used in place of the steps presented in Fig. 4.

We have introduced SPD as our skill model. In this work, they consist of pose means and variances, velocities and accelerations. We utilize the KLfD pipeline to learn the SPD. However, there can be other uses of SPDs such as constructing them by hand for certain robot behaviors or animation (in place of traditional keyframes).

8 Conclusion

We present a Learning from Demonstration framework that accommodates two types of novel input demonstrations, keyframe and hybrid demonstrations, along with traditional trajectory demonstrations. The usability and utility of these novel types were motivated in our previous work. In this article we present methods for learning from all three types of demonstrations in a unified framework. Our methods are based on converting all types of demonstrations into keyframes. Then, the keyframes are aligned and clustered and the skill is reproduced from the obtained clusters. We call this framework *Keyframe-based LfD* (KLfD) and the resulting skill model as *Sequential Pose Distributions* (SPD).

We demonstrated KLfD performs on par with a conventional LfD technique on trajectory demonstrations. This implies that KLfD is a viable alternative even for conventional demonstration types, while accommodating new demonstrations types. We also showed that KLfD is most powerful when the demonstration type is suited to the particular skill being demonstrated. This highlights the strength of hybrid demonstrations that can be suited for any type of skill.

[4] See *e.g.* `http://www.youtube.com/watch?v=5lVxOKSeYsk`

Acknowledgements This research is supported by NSF CAREER grant IIS-1032254.

References

1. Abbeel P, Ng AY (2004) Apprenticeship learning via inverse reinforcement learning. In: Proceedings of the Twenty-first International Conference on Machine Learning, ACM Press
2. Akgun B, Cakmak M, Wook Yoo J, Thomaz LA (2011) Augmenting kinesthetic teaching with keyframes. In: ICML Workshop on New Developments in Imitation Learning
3. Akgun B, Cakmak M, Wook Yoo J, Thomaz LA (2012) Trajectories and keyframes for kinesthetic teaching: A human-robot interaction perspective. In: ACM/IEEE Intl. Conference on Human-robot interaction (HRI)
4. Amor HB, Berger E, Vogt D, Jun B (2009) Kinesthetic bootstrapping: Teaching motor skills to humanoid robots through physical interaction. Lecture Notes in Computer Science: Advances in Artificial Intelligence 58(3):492–499
5. Argall B, Chernova S, Browning B, Veloso M (2009) A survey of robot learning from demonstration. Robotics and Autonomous Systems 57(5):469–483
6. Billard A, Calinon S, Guenter F (2006) Discriminative and adaptive imitation in uni-manual and bi-manual tasks. Robotics and Autonomous System 54(5):370–384
7. Bitzer S, Howard M, Vijayakumar S (2010) Using dimensionality reduction to exploit constraints in reinforcement learning. In: Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, pp 3219–3225
8. Calinon S, Billard A (2007) What is the teacher's role in robot programming by demonstration? - Toward benchmarks for improved learning. Interaction Studies Special Issue on Psychological Benchmarks in Human-Robot Interaction 8(3)
9. Calinon S, Billard A (2009) Statistical learning by imitation of competing constraints in joint space and task space. Advanced Robotics 23(15):2059–2076

10. Calinon S, Guenter F, Billard A (2007) On learning, representing and generalizing a task in a humanoid robot. IEEE Transactions on Systems, Man and Cybernetics, Part B Special issue on robot learning by observation, demonstration and imitation 37(2):286–298
11. Carrillo H, Lipman D (1988) The multiple sequence alignment problem in biology. SIAM Journal on Applied Mathematics pp 1073–1082
12. Eddy S (1998) Profile hidden markov models. Bioinformatics 14(9):755–763
13. Flash T, Hogan N (1985) The coordination of arm movements: an experimentally confirmed mathematical model. The journal of Neuroscience 5(7):1688–1703
14. Gribovskaya E, Billard A (2009) Learning nonlinear multi-variate motion dynamics for real- time position and orientation control of robotic manipulators. In: Proceedings of IEEE-RAS International Conference on Humanoid Robots
15. Halit C, Capin T (2011) Multiscale motion saliency for keyframe extraction from motion capture sequences. Computer Animation and Virtual Worlds 22(1):3–14
16. Hersch M, Guenter F, Calinon S, Billard A (2008) Dynamical system modulation for robot learning via kinesthetic demonstrations. IEEE Transactions on Robotics 24(6):1463–1467
17. Hsiao K (2006) Imitation learning of whole-body grasps. In: In IEEE/RJS International Conference on Intelligent Robots and Systems (IROS, pp 5657–5662
18. Khansari-Zadeh SM, Billard A (2011) Learning Stable Non-Linear Dynamical Systems with Gaussian Mixture Models. IEEE Transaction on Robotics
19. Liu Y, Zhou F, Liu W, De la Torre F, Liu Y (2010) Unsupervised summarization of rushes videos. In: Proceedings of the international conference on Multimedia, ACM, New York, NY, USA, MM ’10, pp 751–754
20. Lowe D (1987) Three-dimensional object recognition from single two-dimensional images. Artificial intelligence 31(3):355–395
21. Miyamoto H, Schaal S, Gandolfo F, Gomi H, Koike Y, Osu R, Nakano E, Wada Y, Kawato M (1996) A kendama learning robot based on bi-directional theory. Neural Netw 9:1281–1302
22. Nair N, Sreenivas T (2007) Joint decoding of multiple speech patterns for robust speech recognition. In: Automatic Speech Recognition Understanding, 2007. ASRU. IEEE Workshop on, pp 93 –98, DOI 10.1109/ASRU.2007.4430090
23. Parent R (2002) Computer animation: algorithms and techniques. Morgan Kaufmann series in computer graphics and geometric modeling, Morgan Kaufmann
24. Pastor P, Hoffmann H, Asfour T, Schaal S (2009) Learning and generalization of motor skills by learning from demonstration. In: IEEE Intl. Conference on Robotics and Automation
25. Ratliff N, Ziebart B, Peterson K, Bagnell JA, Hebert M, Dey AK, Srinivasa S (2009) Inverse optimal heuristic control for imitation learning. In: Proc. AISTATS, pp 424–431
26. Todorov E, Jordan M (1998) Smoothness maximization along a predefined path accurately predicts the speed profiles of complex arm movements. Journal of Neurophysiology 80(2):696–714
27. Wada Y, Kawato M (1993) A neural network model for arm trajectory formation using forward inverse dynamics models. Neural Networks 6 pp 919–932
28. Weiss A, Igelsboeck J, Calinon S, Billard A, Tscheligi M (2009) Teaching a humanoid: A user study on learning by demonstration with hoap-3. In: Proceedings of the IEEE Symposium on Robot and Human Interactive Communication (RO-MAN), pp 147–152

Author Biographies

Baris Akgun is a Ph.D. candidate in Robotics at the Georgia Institute of Technology. He received his B.Sc. degree in Mechanical Engineering and an M.Sc. degree in Computer Engineering from the Middle East Technical University, Turkey in 2007 and 2010 respectively. He is interested in human-robot interaction, learning from demonstration and machine-learning for robotics.

Maya Cakmak is a Ph.D. candidate in Robotics at the Georgia Institute of Technology. She has a B.Sc. degree in Electrical and Electronics Engineering (2005) and an M.Sc. degree in Computer Engineering (2007) from the Middle East Technical University, Turkey. Her research interests include learning from demonstration, human-robot interaction and interactive machine learning.

Karl Jiang is a Ph.D. candidate in Computer Science at the Georgia Institute of Technology. He received his B.S. in Computer Engineering from the University of Miami in 2008. He is interested in human-robot social interaction and how perceptual signals can improve learning interactions.

Andrea L. Thomaz received a B.Sc. degree from the University of Texas, Austin, in 1999, and M.Sc. and Ph.D. degrees from the Massachusetts Institute of Technology, Cambridge, in 2002 and 2006, respectively. She is an Assistant Professor in the School of Interactive Computing, Georgia Institute of Technology, where she directs the Socially Intelligent Machines Lab. Her work focuses on social robots and socially guided machine learning. She is published in robotics, human-robot interaction, and artificial intelligence.