The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Human-Robot Interaction for Learning and Adaptation of Object Movements

Manuel Mühlig, Michael Gienger, Jochen J. Steil

Abstract— **In this paper we present a new robot control and learning framework. By integrating previously presented as well as new methods, the robot is able to learn an invariant and generic movement representation from a human tutor. We argue that in order to apply such generic representations to new situations and thus create a flexible system, the use of interaction is beneficial. The interaction is based on a kinematically controlled model of a human tutor, which is used as a model-based filter and also for recognizing postures that influence the interaction. In addition, a new movement segmentation scheme is presented that is based on correlating movements by the tutor's hand with the salient objects in the scene. The focus of this paper is on the interactive learning aspects of the system and particular emphasis is given to an experiment in which the humanoid robot ASIMO learns from a human tutor. The system includes extensive generalization capabilities that result from an online adaption of the robot's body schema and the exploitation of inter-trial variance from multiple demonstrations. This enables the robot to reproduce the movement in new situations. For example, a stacking task that the tutor performed one-handed can be executed bi-manually by the robot.**

I. INTRODUCTION

In the field of robotics, one of the main goals is to equip a robot with human-like cognitive abilities that allow it to learn from a human tutor. By achieving this we would make a great leap forward, because it would lead to systems which are open to new tasks and a wide range of users.

One of the key points of such learning abilities is generalization, in the sense that the robot extracts the important information from a demonstrated movement. In recent years, a lot of progress has been made in this field.

The authors of [1], propose to learn and represent movements using Dynamic Movement Primitives (DMPs). With these DMPs, it is possible to dynamically adapt a movement to slight changes in the environment. On a more symbolic level, the system from [2] learns the structure of a complex *pick and place* task and generalization is achieved by representing alternative behaviors in this structure. In [3], even the concept of affordances has been used to teach a robot new tasks and represent them in a generic way. In [4], generalization is mainly achieved by using a probabilistic representation with Hidden Markov Models and learning from multiple repetitions.

M. Mühlig and J. J. Steil are with the Research Institute for Cognition and Robotics, Bielefeld University, 33594 Bielfeld, Germany. {muehlig,jsteil}@cor-lab.uni-bielefeld.de
M. Mühlig and M. Gienger are with the Honda Research Institute Europe, Carl-Legien-Strasse 30, 63073 Offenbach/Main, Germany. michael.gienger@honda-ri.de

Recently, there is also a trend to combine different learning approaches together with interaction with a human tutor. A cognitive architecture for this purpose is presented in [5]. However, learning is performed in an offline *Programming by Demonstration* approach. In [6] a framework for teaching a robotic system sequences of picking and placing objects is presented. A tutor uses pointing gestures and speech to guide the robot through the task. Also the authors of [7] exploit such social cues to speed up the learning process within a probabilistic imitation learning framework.

However, there is still something missing for true flexibility. A robot needs a way to apply its generic movement representation to specific situations. This means that the additional degrees of freedom achieved through generalization need to be bound to the current situation.

In this paper we propose a way to bridge this gap. We present a new framework for robot control and imitation learning. This framework integrates previous work as well as new elements that allow a humanoid robot to learn and generalize movement tasks from demonstrations of a human tutor. To apply its learned tasks to specific situations, the robot exploits the interaction with the tutor. This results in a very flexible system where the tutor is not just a passive observer of the robot's actions, but is able to actively guide the robot. The focus of this paper is on the interactive elements and how they are used to bind degrees of freedom of the system to specific situations. In the directly related paper [8], new methods for achieving these degrees of freedom by generalization and invariance are presented.

The remainder of this paper is organized as follows. Section II provides an overview of the complete framework and explains how the elements work together. Interaction is a central element that influences both, the learning process and the movement reproduction. How, is explained in Section III. We emphasize the importance of the interaction by presenting experiments in Section IV. It is shown how the robot interacts with the tutor in order to learn a new movement task and apply it to different situations. We conclude the paper in Section V.

II. ROBOT CONTROL AND LEARNING FRAMEWORK

The framework presented in this paper is depicted in Figure 1 and consists of three hierarchical layers with modules grouped into a perception and a control side as well as interaction modules as central elements. Within this section we explain the framework layer-wise from the bottom to the top.

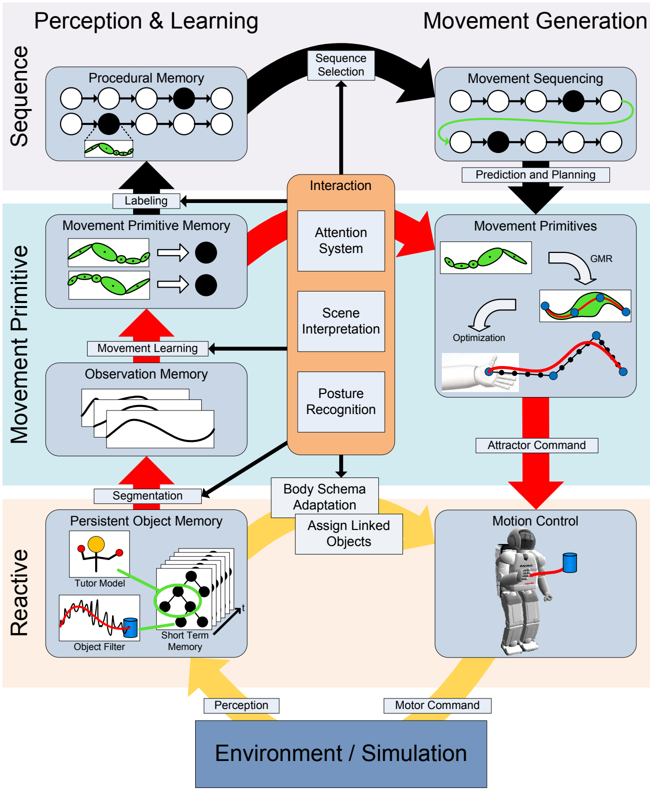

Fig. 1. Structure of the robot control and imitation learning framework

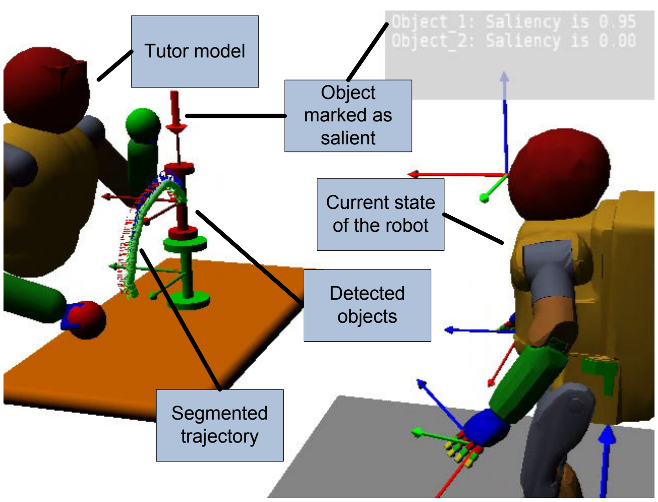

Fig. 2. 3D visualization of the *Persistent Object Memory*

A. Reactive Layer

The bottom layer corresponds to a reactive control system. Information received from simulation or the on-board sensors of the robot are processed in the *Persistent Object Memory*. It can be seen as the robot's perceptual interface to the world. All sensory input of the robot is subsumed in this memory and filtered consistently using a mixture of low-pass, median and model-based filters (for the latter, see also Section III-A). In the concrete case of the experiments, presented in Section IV, the input data comes from the ego-motion-compensated, on-board stereo vision system of the robot.

As the name suggests, the main task of the *Persistent Object Memory* (Figure 2) is to handle information about detected objects. The shapes of the objects are retrieved from a database of known objects and matched to the perceived input. In addition, a confidence value is associated to each object that depends on when the object was last observed. It is used to increase the robustness of the system.

The objects are represented together with the robot's body parts within one single kinematic tree. This makes it possible to define controllers for the robot that operate directly on observed objects. The motion control is based on an inverse kinematic control scheme presented in [9] and based on [10]. This control scheme already allows a very flexible task-level control of the robot by representing movements within egocentric or allocentric frames of reference. But even more, there are two enhancements, presented in the related paper [8], that further increase the flexibility.

On the one hand, the involved body schema of the robot (i.e., the kinematic configuration) can be reconfigured online. This is especially useful for tasks such as grasping, where after a successful grasp the object is attached to the robot's effector. The object can then be controlled like if it is a part of the robot and any optimization and planning process takes this into account.

On the other hand, the concept of linked objects is introduced. They are used to decouple the task descriptors from physical entities (i.e., specific object identities) in order to decrease the number of required task descriptors. By modifying the reference of a linked object, it is possible to achieve different robot behaviors without switching the controlled task.

B. Movement Primitive Layer

On top of the reactive layer, imitation learning capabilities are achieved. This learning is based on previously developed methods, presented in [11], which will be explained briefly within this section. The input of the learning system however is based on a new interaction-based segmentation method that results in object trajectories demonstrated by a human tutor. This segmentation is described in Section III-C.

In the first step, the information coming from several demonstrations of the same task is projected into the task space in which the movement should be learned. Although we already presented an approach to choose such task spaces automatically [12], we simplify the scheme for the experiments in Section IV by selecting the task space manually. The use of task spaces is beneficial, because it accounts for an invariance in the movement representation (e.g., tasks described in relative frames of reference can be executed at various absolute positions). The demonstrated trajectories, represented in task space, are stored in the *Observation Memory* of the framework.

After the different demonstrations have been acquired, we need to account for temporal distortions. This is done by applying Dynamic Time Warping, which results in meaningful spatial variance information that can be exploited later during the movement reproduction. To encode the mean and covariance information of the task we apply multivariate Gaussian Mixture Models. They are trained using a common Expectation-Maximization algorithm with K-Means initialization, leading to a compact, probabilistic representation, which is stored as a movement primitive within the so-called *Movement Primitive Memory*.

On the control side, these movement primitives can be used to initialize and trigger an attractor-based movement optimization that adapts the movement to a new situation. This is necessary, because the movement representation is situation-independent and does not account for robot-specific constraints (e.g., joint limits, self-balance) or environmental constraints (e.g., collisions). To handle these aspects, we incorporate a gradient-based trajectory optimization scheme which has been already presented in [13]. It operates on an attractor-based trajectory generation that describes the task space trajectories with linear attractor dynamics. These dynamics command the motion control system on the reactive layer. In principle, the sequence of attractor vectors is optimized so that cost functions, corresponding to the above mentioned constraints, are minimized. Besides the cost functions for collision avoidance, joint-limit avoidance and so on, a similarity criterion is incorporated that penalizes deviations from the learned movement, but with respect to the variance information.

C. Sequence Layer

To achieve complex tasks it is not sufficient to control the robot based on movement primitives only. The sequence layer therefore allows to combine learned as well as predefined movement primitives into complex sequences. The movement primitives are interconnected with the help of transitions that are triggered by internal (e.g., a robot movement converged to a given target) or external events (e.g., the tutor raised the hand as a stop signal). Such sequences ease the modeling of complex movement chains and the augmentation with learned movements. Furthermore, the system is able to predict and plan across the movement primitives in such chains and command them sequentially to the lower layers.

III. SCENE INTERPRETATION AND INTERACTION

In the previous section we presented an overview of the robot control and learning framework. This framework combines various methods that increase the invariance and generalization capabilities of the system. When the robot is asked to execute a learned task, these additional degrees of freedom need to be bound to the specific situation. We achieve this by including interactive aspects as central elements in our framework. This creates a very flexible system, because the tutor can shape the situation in natural interaction with the robot.

A. Tutor Model

We assume that the robot normally interacts with a human tutor. Therefore, the *Persistent Object Memory* also includes a model of the upper body of a human tutor. The model is controlled in task space (e.g., end-effector positions) using an inverse kinematics control scheme based on [14]. For the experiment presented in Section IV a skin color detector is used to detect the positions of the hands and the head of the tutor. This input is sufficient to control the model in a 9-dimensional task space spanned by the Cartesian coordinates.

In our framework, the tutor model fulfills two tasks. First, it is used as a model-based filter for the hands of the tutor. Joint limits and joint speed limits prevent the body parts of the tutor from moving unnaturally fast. Therefore, movements are interpolated more realistically during phases where the input is missing (e.g., because of occlusions). Second, the model is used for recognizing postures (Section III-B), which can trigger special transitions on the *Sequence Layer*.

But, such a kinematically controlled tutor model can be additionally useful. A common problem when relying on vision input is the detection of hand orientations when grasping objects. The hand is hidden behind the object, which usually leads to a wrong estimation of the tutor's pose. The problem can be solved by aligning the grasp axis of the tutor's hands with the object's major axis if the hand and the object are close together. Two additional dimensions per hand are then added to the task space to control the polar angles of the hands (see [15] for the two-dimensional hand attitude control). This results in a better estimation of the posture.

Furthermore, a tutor model allows the prediction of internal states of the tutor. In our previous work [12] we already showed that by defining cost functions, such as *effort* (torque-based) or *discomfort* (based on joint ranges), we are able to determine which elements of a movement demonstration are important and which just result from the natural posture. This work is based on findings about the mirror system in humans, which claim that we employ our own motor system for recognizing actions of other humans (see also [16]).

B. Posture Recognition

Based on the tutor model, presented in the previous section, we use posture recognition to structure the interaction and for the actual communication with the robot. It is especially used to trigger some transitions within the sequences on the top layer of our framework. As an example, in our experiments described later we used postures like lifting one or both hands in order to command the robot to execute movements with one or both hands. The same postures were used to signalize the robot to remember or forget a demonstrated movement task. By using postures to define how a task should be executed (e.g., one-handed or bi-manual), the invariance gained from the generic task representation within task spaces is directly transformed into flexibility of the whole system. Such postures are recognized by continuously evaluating the positions of the hands relative to the head of the tutor.

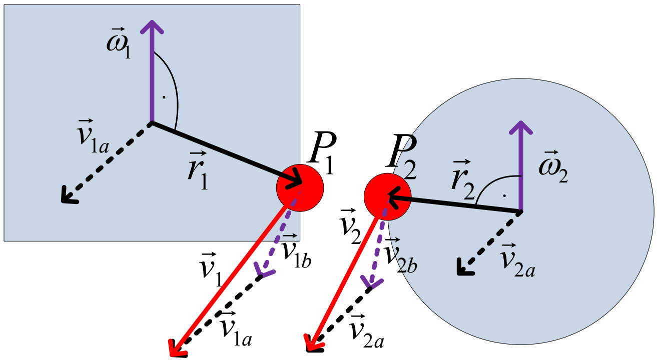

Fig. 3. Calculating the velocities of the nearest points of two objects

C. Movement Segmentation

In section II-B it was already mentioned that the input of the imitation learning system comprises a new segmentation method. We exploit the assumption of an interactive scenario with a human tutor in order to recognize when significant object-related actions are performed.

We propose a new movement segmentation that is based on correlative features between the tutor's hand and objects. The basic idea is that if an object and a hand are near to each other and starting to move with the same velocity, the object is most likely in the hand of the tutor and she/he is manipulating the object actively. This marks the start of a segment. The end of a segment is reached if both conditions become invalid.

To calculate this, first the currently most salient object and the hand are located. The *Persistent Object Memory* holds information about the position as well as the linear and the angular velocity of their center points. In addition, the exact or approximate shapes of the object and the hand are available. It is however insufficient to use the velocities directly. The reason becomes clear for the example of a human tutor manipulating a large stick. The center point of the stick may be far away and moving fast if the tutor grasped the stick on one side and turns her/his hand. Comparing the velocities of the hand and the object would result in a large deviation and not be recognized as moving similarly, which is indeed wrong.

To overcome these problems, the correlation of the object and the hand movement is calculated in the following way. First, the nearest points P_1 and P_2 for the object and the hand are calculated by taking the shapes of both into account (see Figure 3). The correlation of their velocities is now calculated for these points instead of the center points of the object and the hand. For each point P_i, the velocity $\mathbf{v}_i$ is the sum of the linear velocity $\mathbf{v}_{ia}$ and the outer product of the angular velocity $\boldsymbol{\omega}_i$ and the radius of the point $\mathbf{r}_i$ (see Equation 1).

$$\mathbf{v}_i = \mathbf{v}_{ia} + \boldsymbol{\omega}_i \times \mathbf{r}_i \tag{1}$$

Now the correlation function f that consists of two terms is calculated continuously (Equation 2). The vectors $\mathbf{p}_1$ and $\mathbf{p}_2$ relate to the position of P_1 and P_2, respectively.

$$f(\mathbf{p}_1, \mathbf{p}_2, \mathbf{v}_1, \mathbf{v}_2) = \frac{1}{2} \cdot f_1(|\mathbf{p}_1 - \mathbf{p}_2|) + \frac{1}{2} \cdot f_2(\mathbf{v}_1, \mathbf{v}_2) \tag{2}$$

The value of the first term f_1 depends on the distance d between the two points (Equation 3). It switches softly from Zero to One near the threshold c_2, thus signalizing that the hand is near the object. The switching is done by using the Sigmoid function from Equation 4.

$$f_1(d) = \varsigma(c_1(d - c_2)) \tag{3}$$

$$\varsigma(x) = \frac{1}{1 + e^{-x}} \tag{4}$$

The second term f_2 is similar to f_1 but depends on the velocities of both points.

$$f_2(\mathbf{v}_1, \mathbf{v}_2) = \alpha(\mathbf{v}_1, \mathbf{v}_2) \cdot \varsigma(c_3(|\mathbf{v}_1 - \mathbf{v}_2| - c_4)) \tag{5}$$

The value of f_2 increases if the difference of the velocities of P_1 and P_2 is large, but only if the function α is larger than Zero. This function depends on the absolute velocities of both points in order to allow function f_2 to contribute to f only if the hand and the object are moving at all.

$$\alpha(\mathbf{v}_1, \mathbf{v}_2) = \varsigma(c_5(|\mathbf{v}_1| - c_6)) \cdot \varsigma(c_7(|\mathbf{v}_2| - c_8)) \tag{6}$$

The values of function f during the segmentation follow a trimodal distribution and can be tuned using the constants c_i. If the hand and the object are not moving or are not near each other, then f becomes a value near Zero. If the hand and the object are together, but not or only slightly moving, f is around 0.5. If now the hand and the object are close and their nearest points are moving into similar directions with similar velocity the value of f rises towards One. This behavior is very advantageous for the segmentation of distinct object movements by simply applying two thresholds. The experiments in Section IV illustrate this in more detail. For these experiments, the values of the constants and the thresholds were chosen a priori.

D. Attention and Gazing

For human-robot interaction, an attention system and the robot's gazing behavior are essential elements. On the one hand they allow the tutor to highlight important scene elements (e.g., the objects that are involved in a task demonstration). On the other hand the robot gives feedback to the tutor by gazing at what it "believes" to be important.

Usual attention mechanisms combine bottom-up and top-down processing of sensory data (e.g., color images and depth maps). Such approaches are presented in [17] and [18].

We apply a more object-related attention mechanism. In our framework, a saliency value is associated with each detected object. This saliency value has a temporal decay and can be increased either by moving or shaking objects or by pointing at them. Internally, a list of all objects is maintained, sorted according to their saliency values and with a hysteresis applied.

These saliency values define the gazing behavior of the robot. A virtual gazing point is calculated according to the confidence and saliency values of the detected objects. The position of the gazing point $\mathbf{p}_g$ is calculated according to Equation 7. The vector $\mathbf{p}_i$ corresponds to the position of object i and scalar s_i to its saliency value. Only objects with a confidence $conf_i$ that is high enough (larger than constant c_s) are involved in the calculation.

$$\mathbf{p}_g = \sum_i \frac{\mathbf{p}_i s_i w_i}{\sum_i s_i w_i} \quad \forall i : conf_i > c_s \tag{7}$$

Additionally the factor w_i is included that increases the importance of objects that are near to the border of the field of view. This leads to a behavior in which the robot tries to keep all important (i.e., recently highlighted) objects in its view. This reactive gazing behavior can be influenced by elements from the top layer of our architecture. For example, if the robot needs a response from the tutor, the weight for the tutor's head w_{head} is increased. The robot then gazes at the tutor and continuously tracks her/his head.

E. Attention and Movement Representation

The attention mechanism also directly influences the learning and reproduction of movements. These movements are represented within task spaces that relate the most salient object to the second-most salient one. This, together with the concept of linked objects, leads to a more general representation of learned movements. In addition, this offers a flexible way to control what the robot should do without changing the underlying movement representation. For example, the robot could have learned to put the most salient object on top of the second-most salient one. By highlighting different objects before asking the robot to execute the movement, the tutor can define which objects are involved.

IV. EXPERIMENTS

In this section we present three real world experiments with the humanoid robot ASIMO and the presented robot control and learning framework. We want to emphasize how the interactive elements in the system, on the one hand help to learn a generic movement representation, but also on the other hand allow to apply such learned movements to specific situations. The first experiment shows the results of the segmentation algorithm from Section III-C and the second experiment demonstrates the completely interactive teaching of a movement to the robot. The third experiment shows how the robot can exploit learned variance information to adapt its movement to new situations. The setup (see Figure 4 or 5) is the same for all experiments. ASIMO is standing half a meter away from a table on which object-related movements are demonstrated. The robot is observing the scene with its on-board stereo vision system. Color-based segmentation allows it to track the 3D position and 1D orientation of the red and the green object as well as the head and the two hands of the tutor. The whole interaction is modeled using a complex sequence of movement primitives and transitions on the highest layer of our framework.

A. Segmentation

Figure 4 shows the principle of the segmentation based on the hand-object correlations. One can see that the demonstration can be segmented by simply applying two thresholds for the start and the end, respectively. When the tutor grasps the object the value of f (Equation 2) rises to about 0.5, not yet starting a segment. Then, the object is moved and the term of Equation 5 contributes to the value, increasing it to above the *start threshold*. The segment endures until the tutor finally releases the object, because this leads to the value falling below the *end threshold*.

The proposed method provides a good way to segment object-related movement in a natural way. Furthermore, the assumptions about the hand-object relations reduce the probability of over-segmentation.

B. Typical Tutoring Scenario

In the second experiment a typical interaction cycle during the teaching of a robot is presented. For this experiment, the task of the robot was to learn to put one object on top of another. The robot is able to learn and generalize this to a new situation. Although the movement was seen performed with one hand only, the robot is able to fulfill the request of the tutor to reproduce it with two hands. Figure 5 shows snapshots of the scene during the interaction and a qualitative analysis of the human-robot interaction, highlighting the interplay between internal elements.

In the beginning, the tutor catches the attention of the robot by tapping on the object that will be involved in the next steps. The robot recognizes this and changes its gazing behavior, which indeed is a hint for the tutor that the robot is now attentive. The tutor starts to demonstrate the task, which the robot internally segments using the mechanism of the previous experiment. After the robot recognized the end of the demonstration it is gazing at the tutor's face and awaiting a response. By raising his left hand, the tutor confirms that this was a demonstration of interest and the robot should learn it. The robot recognizes this and reverts to its normal gazing behavior. To abbreviate this example, the tutor demonstrates only once, but it is possible to repeat the previous steps with additional demonstrations.

After the demonstration, the tutor puts the objects on the robot's side of the table and asks the robot to learn and reproduce the task with two hands by raising both hands. This starts several processes in parallel. First, the representation of the objects in the *Persistent Object Memory* is being frozen. This is necessary because the robot is not able to see the objects during manipulation. Second, the learning process is started, which performs the learning steps described in Section II-B. Third, the robot starts walking towards the table and grasps the objects. Note that the learning is done in parallel with the robot's walking and grasping movements. This is also true for the prediction of the future state when the robot will have both objects in its hands and for the optimization of the movement from this state on.

Sometimes it may happen that the robot predicts that the movement is too difficult to be executed properly. This may result from a predicted violation of joint limits or collisions. In our example we show such a case. After the robot grasped the second object, it gazes at the tutor's face and tells him verbally that the movement may be too difficult. The tutor

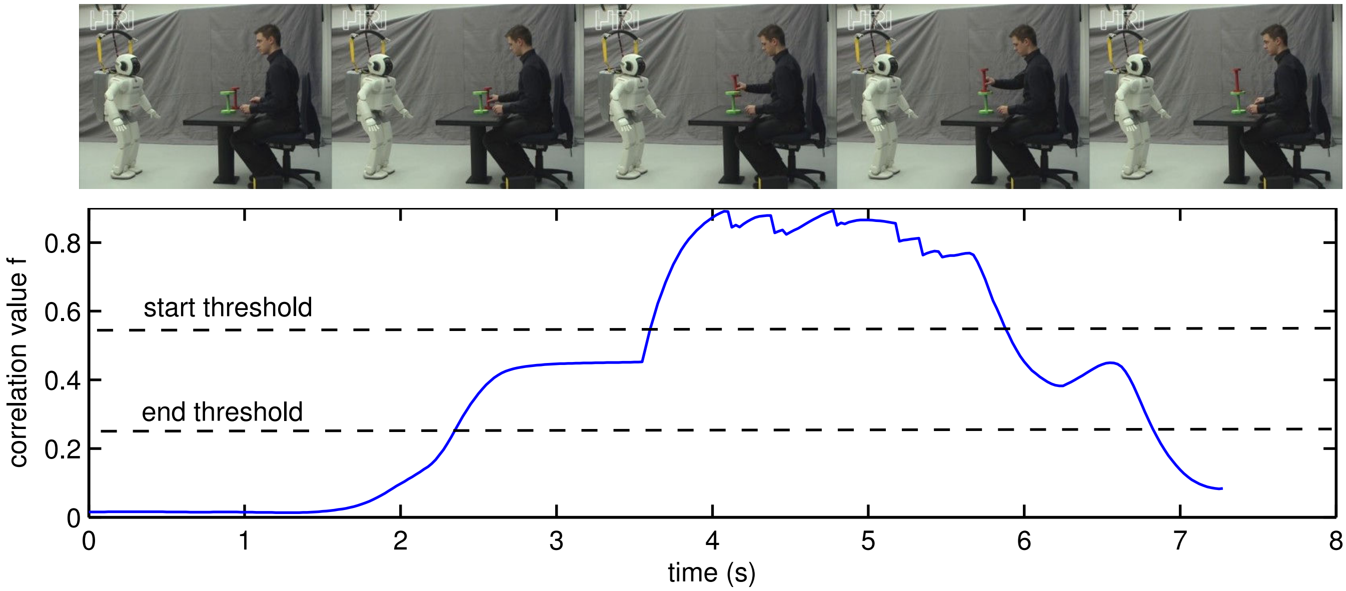

Fig. 4. Plot of the correlation value that is used for segmenting object-related movements

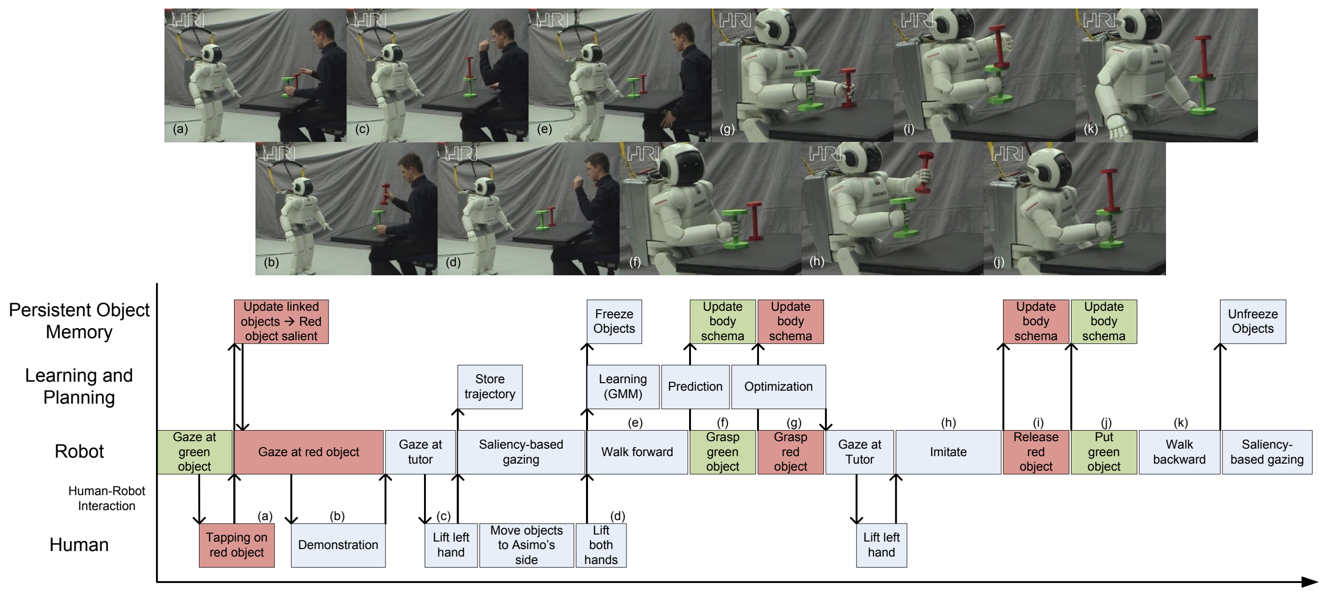

Fig. 5. Illustration of the interaction and the interplay between internal elements during an experiment

now has to decide if the robot should try it anyway or refrain from it. In our example the tutor raises his left hand meaning the former. The robot now reproduces the task successfully, puts the objects onto the table and retreats from the table. The objects are unfrozen again and the robot can engage in further interaction. Note that during the grasping and the releasing of objects the body schema of the robot changes too. This is one of the generalization features that allows the robot to actually perform the movement bi-manually, although it has seen the demonstration only one-handed.

C. Exploitation of Variances

In another experiment (Figure 6), the robot is asked to reproduce the same movement with one hand. This is done in two different situations. Firstly, without any obstacle and secondly with a yellow box blocking the direct path of the red object. In both cases the tutor highlighted the green and the red object in before, so that the generic representation of the stacking movement is applied to those two. The movement itself was demonstrated multiple times instead of only once. This leads to more variance in the demonstration. During the reproduction, this variance is exploited by the robot to avoid a collision with the yellow box. The figure shows that the robot is still able to fulfill the task. In fact, the experiment shows that generalization is not only achieved by learning the task in object-related task spaces, but also by using the probabilistic representation with Gaussian Mixture Models. This is explained more detailed in [11].

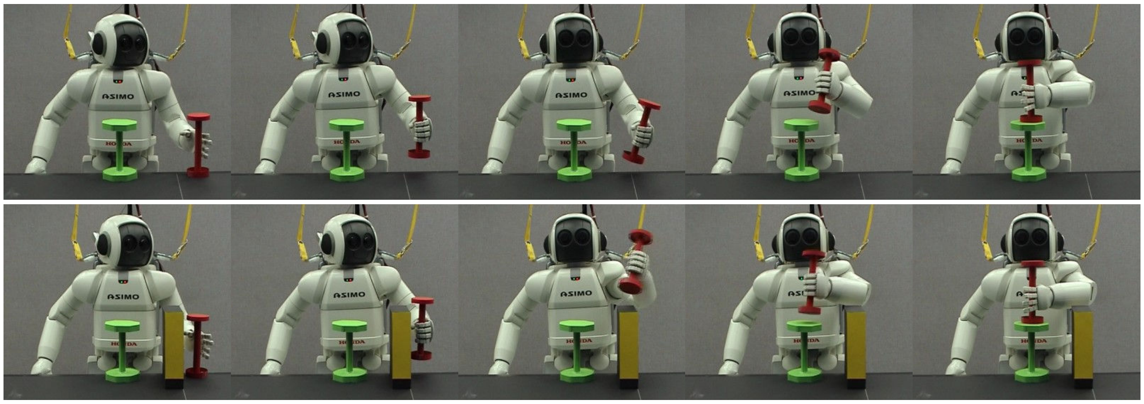

Fig. 6. One-handed imitation with and without obstacle

V. CONCLUSION

We presented a framework that allows a robot to learn and reproduce movement tasks in interaction with a human tutor. This interaction is important in two ways. On the one hand, we use it to generate degrees of freedom in the movement representation, which improves the generalization capabilities of the robot. On the other hand, it is used during the movement reproduction to map the degrees of freedom to specific situations. In particular, our experiments show that interaction leads to flexibility in the following ways:

- Saliency and the robot's attention are used to determine the objects from which the robot should learn.
- The same features are used to define which objects should be manipulated by the robot.
- The tutor's postures "tell" the robot how to reproduce a learned task (e.g., one-handed or bi-manually).
- By introducing variance into the demonstrations, the tutor implicitly allows the robot to avoid obstacles and still perform the task.
- If there is uncertainty about the correct way to reproduce a movement, the robot can verbally ask the tutor for a decision.

In addition to the interactive elements, our framework incorporates a flexible robot control approach that allows to define very complex task spaces. This, in turn, allows to learn tasks as generic representations based on object relations. Furthermore, the framework includes online body schema adaptation and the concept of linked objects, which increase the generalization capabilities of the system even more.

In this paper, we particularly focused on interaction to achieve flexibility. In future, we will investigate this further, but also try to increase the autonomy of the system, for example by including higher-level planning approaches.

REFERENCES

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

[2] M. Nicolescu and M. J. Mataric, "Task learning through imitation and human-robot interaction," *Models and Mechanisms of Imitation and Social Learning in Robots, Humans and Animals: Behavioural, Social and Communicative Dimensions*, pp. 407–424, 2006.

[3] M. Lopes, F. S. Melo, and L. Montesano, "Affordance-based imitation learning in robots," in *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2007.

[4] T. Inamura, I. Toshima, H. Tanie, and Y. Nakamura, "Embodied symbol emergence based on mimesis theory," *The International Journal of Robotics Research*, vol. 23, pp. 363–377, 2004.

[5] C. Burghart, R. Mikut, R. Stiefelhagen, T. Asfour, H. Holzapfel, P. Steinhaus, and R. Dillmann, "A cognitive architecture for a humanoid robot: A first approach," in *Proceedings of 2005 5th IEEE-RAS International Conference on Humanoid Robots*, 2005, pp. 357–362.

[6] P. Mcguire, J. Fritsch, J. J. Steil, F. Röthling, G. A. Fink, S. Wachsmuth, G. Sagerer, and H. Ritter, "Multi-modal human-machine communication for instructing robot grasping tasks," in *In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS*, 2002, pp. 1082–1088.

[7] S. Calinon and A. Billard, "A framework integrating statistical and social cues to teach a humanoid robot new skills," in *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA), Workshop on Social Interaction with Intelligent Indoor Robots*, May 2008.

[8] M. Gienger, M. Mühlig, and J. J. Steil, "Imitating object movement skills with robots – a task-level approach exploiting generalization and invariance," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010.

[9] M. Gienger, H. Janssen, and C. Goerick, "Task-oriented whole body motion for humanoid robots," in *2005 5th IEEE-RAS International Conference on Humanoid Robots*, Dec. 2005, pp. 238– 244.

[10] A. Liégeois, "Automatic supervisory control of configuration and behavior of multibody mechanisms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 7, no. 12, pp. 861–871, 1977.

[11] M. Mühlig, M. Gienger, S. Hellbach, J. J. Steil, and C. Goerick, "Task-level imitation learning using variance-based movement optimization," in *IEEE International Conference on Robotics and Automation (ICRA 2009)*, May 2009.

[12] M. Mühlig, M. Gienger, J. J. Steil, and C. Goerick, "Automatic selection of task spaces for imitation learning," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009.

[13] M. Toussaint, M. Gienger, and C. Goerick, "Optimization of sequential attractor-based movement for compact behaviour generation," in *7th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2007)*, 2007.

[14] Y. Nakamura, *Advanced Robotics: Redundancy and Optimization*. Addison-Wesley Publishing Company, 1991.

[15] M. Gienger, H. Janssen, and C. Goerick, "Exploiting task intervals for whole body robot control," in *Proceedings of the International Conference on Intelligent Robots and Systems*, 2006.

[16] M. J. Mataric and M. Pomplun, "Fixation behavior in observation and imitation of human movement," *Cognitive Brain Research*, vol. 7, pp. 191–202, 1998.

[17] Y. Nagai, C. Muhl, and K. J. Rohlfing, "Toward designing a robot that learns actions from parental demonstrations," in *2008 IEEE International Conference on Robotics and Automation*, Pasadena, CA, USA, 19/05/2008 2008, pp. 3545–3550.

[18] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.