Autonomous Robots manuscript No.
(will be inserted by the editor)

Interactive Imitation Learning of Object Movement Skills

Manuel Mühlig · Michael Gienger · Jochen J. Steil

Received: date / Accepted: date

Abstract In this paper we present a new robot control and learning system that allows a humanoid robot to extend its movement repertoire by learning from a human tutor. The focus is learning and imitating motor skills to move and position objects. We concentrate on two major aspects. First, the presented teaching and imitation scenario is fully interactive. A human tutor can teach the robot which is in turn able to integrate newly learned skills into different movement sequences online. Second, we combine a number of novel concepts to enhance the flexibility and generalization capabilities of the system. Generalization to new tasks is obtained by decoupling the learned movements from the robot's embodiment using a task space representation. It is chosen automatically from a commonly used task space pool. The movement descriptions are further decoupled from specific object instances by formulating them with respect to so-called linked objects. They act as references and can interactively be bound to real objects. When executing a learned task, a flexible kinematic description allows to change the robot's body schema online and thereby apply the learned movement relative to different body parts or new objects. An efficient optimization scheme adapts movements to such situations performing online obstacle and self-collision avoidance. Finally, all described processes are combined within a comprehensive architecture. To demonstrate the generalization capabilities we show experiments where the robot performs a movement bimanually in different environments, although the task was demonstrated by the tutor only one-handed.

Keywords Imitation Learning · Human-Robot Interaction · Robot Control · Kinematics

Manuel Mühlig · Michael Gienger
Honda Research Institute Europe
Carl-Legien-Strasse 30, 63073 Offenbach/Main, Germany
E-mail: {manuel.muehlig,michael.gienger}@honda-ri.de

Jochen J. Steil
Research Institute for Cognition and Robotics (CoR-Lab)
Bielefeld University, 33615 Bielefeld, Germany
E-mail: jsteil@cor-lab.uni-bielefeld.de

1 Introduction

One of the key abilities of a cognitive robotic system is to extend its own movement repertoire by learning new skills from humans. While in the early days of this research field this meant mimicking movements of a tutor, apparently this is not enough to improve the abilities of a robot. More recently, work has consequently turned towards researching methods for learning and representing movements with the goal of generalization. The robot's target then is to learn the important and invariant aspects of movements and to apply this knowledge to new situations.

One way of learning these important aspects is the use of probabilistic methods. For example, Asfour et al. (2006) present a method to learn arm movements from a human tutor. Features that are invariant in multiple demonstrations of the same task are recognized and learned with Hidden Markov Models (HMMs). The movement is encoded using joint angles and the position and orientation of the Tool Center Point (TCP). Also Calinon and Billard (2008) present a system to learn a probabilistic representation of a demonstrated task. There, the variance information encoded in Gaussian Mixture Models (GMMs) relates to different constraints of the task. Furthermore, the probabilistic representation allows to integrate social cues, such as speech and gaze to scaffold the learning process. Eppner et al. (2009) use Dynamic Bayesian Networks (DBNs) as more general probabilistic

models. In their system, the variances of task-level and joint-level policies determine how accurate a motion should be tracked by the robot. Another interesting approach for learning object movements in 2D space using HMMs is presented by Sugiura et al. (2010). Like us, they are one of the few groups that include a selection of feasible reference frames in their movement learning process.

Instead of encoding movement alternatives probabilistically, Ijspeert et al. (2003); Schaal et al. (2003); Pastor et al. (2009) use Dynamic Movement Primitives (DMPs) to achieve robustness against spatial and temporal disturbances. The efforts of Khansari-Zadeh and Billard (2010) extend this work by combining a probabilistic representation and DMPs with an optimization approach called SEDS. This optimization allows to counteract possible instabilities of the dynamical system in regions not covered by the demonstrations. Kormushev et al. (2010) use information from the kinesthetic demonstrations of a pancake flipping task together with the PoWER reinforcement learning approach introduced by Kober and Peters (2009). With this, not only movement trajectories are reproduced but also the stiffness parameters of the robot can be adapted to successfully reproduce the task.

A common difficulty for teaching movements to a robot is the correspondence problem, which is the transfer of movement skills to different embodiments. While kinesthetic teaching circumvents this problem to some extent, it is not applicable to every robot. Lopes and Santos-Victor (2005) for instance address this problem by projecting the observed movements into the tutor's frame of reference using a viewpoint transformation. However, movement is represented and reproduced using a fixed mapping of visual input to the degrees of freedom of the system. For mapping movements between different body configurations Acosta-Calderon and Hu (2005); Hersch et al. (2008); Stoytchev (2003); Nabeshima et al. (2006) investigated the concepts of body schema and body percept. In Azad et al. (2007) the problem is addressed with an intermediate kinematic model, the so-called Master Motor Map.

For complex behavior of a robot it is not sufficient to learn individual movement skills alone. They need to be combined and sequentialized. For this, it seems necessary to bridge the gap to a more symbolic representation. Inamura et al. (2004) propose a probabilistic framework based on Hidden Markov Models. Full body movement skills are learned from a human tutor and a symbolic representation called proto symbols emerges. Having such a symbolic representation, methods and frameworks like presented in Beetz et al. (2010); Nicolescu and Matarić (2006) can be used for planning more complex movement sequences. Another possibility for abstracting movements to higher-level primitives is proposed by Lopes et al. (2007). They use the concept of affordances to build up a general world model in which effects of actions can be observed by the robot. The robot therefore can not only learn the movement itself, but also the effects that it should achieve.

For organizing learned and predefined movement skills, Burghart et al. (2005) suggest a state machine approach. Tasks are divided into subtasks that have a goal and possible error states. Communication with the human is used to solve errors. An alternative to classical state machines is presented by Toussaint et al. (2010). They propose a probabilistic framework to combine movement and trajectory planning with higher-level symbolic reasoning. Yamashita and Tani (2008) are pursuing a different approach. They present a neural network representation in which a functional hierarchy of movement primitives emerges automatically. However, their focus is more on biologically plausible models instead of a real-time interaction between tutor and robot.

As learning always involves interaction between the robot and the human tutor, most learning architectures employ methods that allow a dialog between both. A hybrid architecture to instruct a robot in grasping tasks has been proposed by McGuire et al. (2002); Steil et al. (2004). They incorporate active vision, gestural instruction and a dialog system and couple these elements with a hierarchical movement generation system. Bohg et al. (2009) also present a comprehensive architecture that includes higher-level symbolic reasoning. Focus of their work is grasp-oriented visual perception where they also combine visual attention and different visual cues with grasp planning and inference strategies.

Although many existing frameworks comprise sophisticated methods for movement learning and robot control, their focus is usually very specific. They only focus on single aspects and postpone the investigation of a whole systems approach. We try to broaden the view by efficiently combining multiple generalization concepts into one framework. With this we give a robot the capability to learn as much as possible from the tutor's demonstrations and to perform the acquired movement skill in different situations.

The architecture presented in this paper is based on prior work in the area of imitation learning (Mühlig et al. 2009a,b), movement control (Gienger et al. 2005), and optimization (Toussaint et al. 2007). It follows up on our recent publications (Gienger et al. 2010a; Mühlig et al. 2010) and extends them by providing a more thorough description of the system, by enhancing the object representation with task-relevant feature points

and by incorporating a mechanism to determine the most appropriate task space into the learning process. We will focus on learning and imitating object movement skills, and concentrate on two major aspects in this work: First, the presented system allows a human tutor to teach a robot new object movement skills in interaction, and to instruct it to imitate them in a variety of different styles with varying objects in changing scenarios. Second, we combine several concepts for generalization:

- We address the correspondence problem of transferring the demonstrated movement skills to different embodiments using a flexible task space representation.
- Learned movement skills can be applied to different situations using an efficient optimization scheme that exploits probabilistic information.
- A flexible kinematic description allows to imitate the learned skills in different styles by changing the robot's body schema online.
- Learned movements generalize over different objects due to an object feature point representation and the automatic selection of feasible task spaces.

The remainder of this paper is organized as follows. In Section 2 we begin with a structural overview of the learning and control architecture. The subsequent sections explain the elements in more detail. Section 3 presents the perceptual elements of the architecture, explains how the scene is interpreted by the system and how interaction can be used to influence this interpretation. Based on this scene representation, we perform movement segmentation, task space selection as well as movement learning, all explained in Section 4.

Subsequently, Section 5 presents how movement primitives are sequentialized, optimized and executed by the robot. Finally, in Section 6 we show two experiments to highlight the key abilities of the presented architecture. The paper is concluded with a discussion and a brief outlook in Section 7.

2 System outline

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. Additionally, the framework includes interaction modules as central elements. Within this section we briefly summarize the framework layer-wise from bottom to top. The subsequent sections elaborate on the individual elements in more detail.

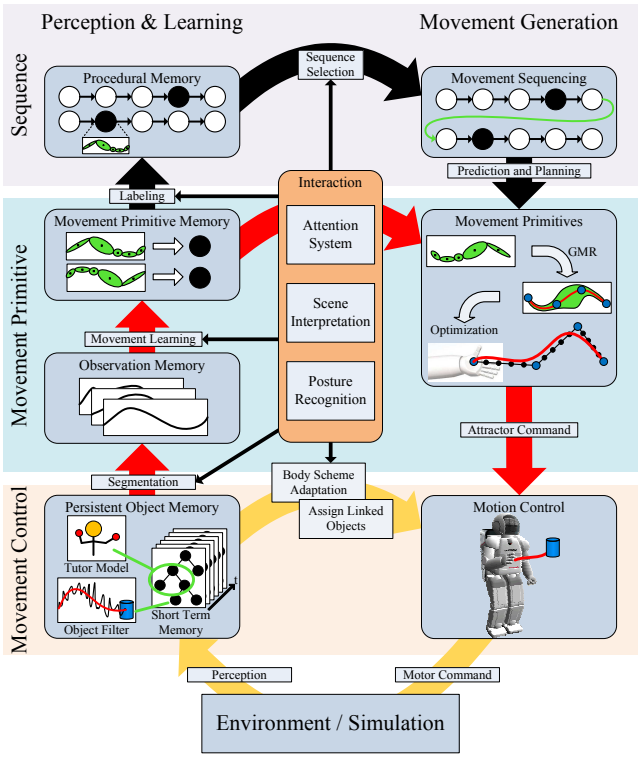

Fig. 1 System architecture overview of the presented robot control and imitation learning framework.

2.1 Movement Control Layer

The bottom layer corresponds to a movement control system. Information received from simulation or the on-board sensors of the robot are processed in the *Persistent Object Memory* (see Section 3.1). For the experiments presented in Section 6, the input data is provided by the ego-motion-compensated, on-board stereo vision system of the robot or a magnetic-field-based motion tracking system.

The information on objects as well as the robot's body parts are represented within one single kinematic tree. This makes it possible to define controllers for the robot that operate directly on observed objects. We incorporate a flexible inverse kinematics control scheme, which allows to define tasks within egocentric or allocentric frames of reference.

2.2 Movement Primitive Layer

On top of the movement control layer, imitation learning capabilities are achieved. Task demonstrations of a tutor can be recognized and segmented automatically using a kinematic model of a human. This segmentation is described in Section 4.1.

In the first step, the acquired trajectories from several demonstrations of the same task are projected into

the task space in which the movement should be learned. While such task spaces are commonly predefined, we employ methods for automatically selecting in which task space a movement is represented best. Selecting an appropriate task space is beneficial since it introduces invariance into the movement representation (e. g., a bimanual task 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*.

The observed trajectories are then learned and stored in form of movement primitives. In the presented framework, the term movement primitive stands for a task-level representation of a goal-directed movement with a defined start and end condition and a fixed set of control (task) variables. These learned movement primitives can be executed reactively or be subject to an optimization procedure that respects additional constraints (e. g., collision avoidance).

2.3 Sequence Layer

To achieve complex tasks it is not sufficient to control the robot based on single movement primitives alone. The sequence layer therefore allows to combine learned as well as predefined movement primitives into complex sequences. The movement primitives are organized in a hierarchical state chart and interconnected by 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). This eases the modelling of complex movement chains and the augmentation with learned movements. The presented system is also able to predict and plan across the movement primitives in such chains and command them sequentially to the lower layers.

2.4 Interaction module

The *Interaction* module plays a central role in the presented framework. Through interaction a tutor can demonstrate movements to the robot and highlight the objects that are important for the task. Further, the tutor can instruct the robot to execute a certain movement sequence and again mark the objects to which this sequence should be applied. Thus, the interaction module influences many elements of the framework such as the *Segmentation*, *Movement Learning*, *Labeling* and *Sequence Selection*.

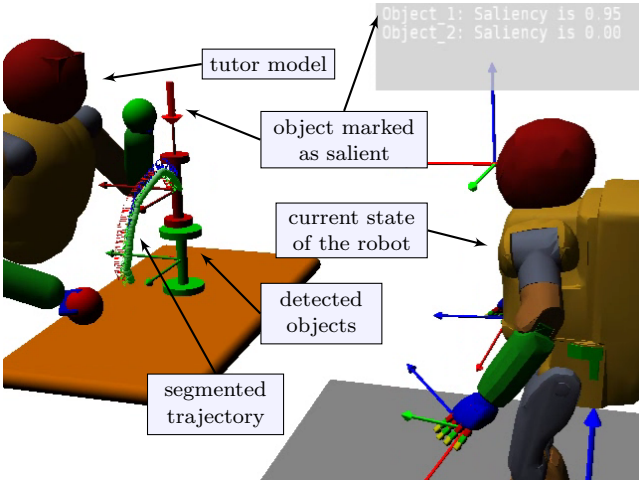

Fig. 2 Visualization of a scenario: The *Persistent Object Memory* comprises the tutor and the objects present in the scene.

3 Perception and interaction cues

An important ability of a cognitive system is the perception and representation of perceived elements. This representation has a variety of functions. It should provide a stable "image" of the world, fuse sensory signals and augment perceived elements with stored information. Further, it should allow flexibility and generalization for the system by representing behavior-relevant information. These points are addressed in this section.

3.1 Persistent object memory

The *Persistent Object Memory* (POM) is the robot's perceptual interface to the world. All sensory information (vision and proprioceptive information) is subsumed and represented in this memory. This representation is based on a kinematic tree of the world including the robot, where all entities can be updated through sensors. These sensory signals are stabilized using a combination of low-pass, median and model-based filters.

When performing experiments with our humanoid robot we can rely on a variety of input processing methods to detect scene elements based on stereo vision only. Planes, like tables and chairs, can be detected using methods such as presented by Heracles et al. (2009). Simple color tracking methods (Bolder et al. 2007) allow to detect uniformly colored objects and skin color blobs. By applying some model-based assumptions also more complex objects like baskets are detectable (Schmuederich 2010). Usually a set of predefined objects is expected to be perceived during an experiment. However, it is also possible to automatically generate new object descriptions and insert them into the kinematic tree as

shown in Einecke et al. (2011). An example for the 3D visualization of the POM is depicted in Figure 2.

The perceived scene can further be supplemented with predefined, static elements such as a fixed table or wall. Furthermore, additional information about object shapes can be incorporated. For example, in the first experiment (see Section 6) the system detected the colored objects using 3D color tracking and augmented the perception with predefined shape information.

All sensory inputs are processed in each frame, and are time-synchronized with the robot's proprioceptive sensor data. This allows to compensate the ego-motion of the robot and to account for unreliable perception due to occlusion of the visual field caused by the robot itself. The details of the algorithms are beyond the scope of this paper, please see Schmuedderich (2010) for more details.

As sensor information might be noisy and unreliable the system includes confidence values for each detected object. If an object is not updated by new sensor values, then this confidence decays over time. We apply a Sigmoidal function of the time t that passed since the last sensor update. It is parametrized in such a way that the confidence begins to decay after 5 seconds without sensor update and becomes almost 0.0 after 10 seconds ($a = -2$ and $b = 15$):

$$conf(t) = \varsigma(a \cdot t + b) \quad \text{with} \quad \varsigma(x) = \frac{1}{1 + e^{-x}} \tag{1}$$

The POM further includes a short term memory that stores the complete state information for a short time period (i. e., a few minutes). This allows to perform operations on past experiences like segmentation and extraction of object trajectories.

3.2 Tutor model

We assume that the robot interacts with a tutor. Therefore, the POM also includes a model of the upper part of a human's body. While quite sophisticated approaches have been suggested (see e. g., Hecht et al. 2009), the model has been kept simple in this work. It is an enabling element for the human-robot interaction and has been realized to achieve the required functions. Future work will focus on improving this model. The tutor's movement is computed using an inverse kinematics algorithm based on Nakamura (1991): The 3D position of tutor's hands and head are determined using the stereo camera system employing skin color detection and depth computation. These values are augmented into a 9-dimensional task vector, which is projected on the tutor's pose using kinematic redundancy resolution as in Eq. (13). The camera's field of view and the experimental setup allow to track the tutor's pose while gazing at the most salient objects.

The tutor model fulfills two tasks. First, it serves 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 filtered more realistically during phases where the input is missing (e. g., because of occlusions). Second, the model is used for recognizing simple postures (Section 3.3), which can trigger transitions of the movement primitive state chart on the *Sequence Layer* (Section 5.1).

But, such a tutor model can be additionally useful. A common problem when relying on vision input is the detection of the hand orientation when grasping objects. The hand is often hidden behind the object, which can lead to a wrong estimation of the tutor's pose. Assuming a power grasp, the problem can be solved by aligning the grasp axis of the tutor's hand with the object's major axis if hand and object are close together. This results in a better estimation of the tutor's posture.

Furthermore, a tutor model allows the prediction of internal states of the tutor. In our previous work (Mühlig et al. 2009b) we have shown that by defining cost functions, such as *effort* (torque-based) or *discomfort* (based on joint ranges), it can be possible to determine which elements of a movement demonstration are important and which just result from a natural posture. This work is based on findings about the mirror system in the human brain. It is claimed that humans employ their own motor system for recognizing actions and intentions of others (see also Matarić and Pomplun 1998).

3.3 Posture recognition

Based on the tutor model, basic posture recognition is performed to structure the interaction and communication with the robot. This is mainly used to trigger transitions within the hierarchical state chart on the top layer of the framework. Postures are recognized by continuously evaluating the positions of the hands relative to the head of the tutor. For example, in the experiments described later, postures like raising one or both hands are used to instruct the robot to imitate movements one-handed or bimanually. The same postures are used to indicate the robot to remember or ignore a demonstrated movement.

3.4 Saliency mechanism

For human-robot interaction, an attention system to guide the robot's gazing behavior is essential. In the presented system, it allows the tutor to highlight important scene elements (e. g., the objects that are involved in a task demonstration). On the other hand, the robot is able to give feedback to the tutor by gazing at what it "believes" to be important.

The most common computational approaches to attention combine pixel-wise bottom-up feature channels into saliency maps, from which the maximally activated pixel is selected for gazing (Itti et al. 1998; Nagai et al. 2008).

We apply a more object-related account of attention that has recently been substantiated by growing experimental evidence (Scholl 2001) and also picked up in several computational models (Wischnewski et al. 2010; Orabona et al. 2007; Walther and Koch 2006). However, in accordance with our object-related tasks, we directly assign saliency values s_i only to detected objects. These values have a temporal decay and can be increased either by moving or shaking objects or by a tutor pointing at them. Additionally, a list of all objects is maintained, sorted according to their saliency values. The saliency computation includes a small hysteresis to make reorganizing the list insensitive to sensor noise: A small constant value is added to the more salient object within the sorting algorithm. Since the saliency is defined for each object, regardless if it is visible or out of view, the saliency list can consistently be computed. However, we should note that only the most salient objects are relevant for segmentation and movement imitation.

This saliency list allows quick access to the most salient objects, which is utilized by the concept of *Linked Objects*, explained in the next section.

3.5 Linked objects

A cognitive robotic system naturally operates within dynamic environments in which the number of objects as well as their identities and geometrical shapes are not known in advance. The representation of movement tasks must therefore be decoupled from concrete object identities. This increases the reusability of learned tasks and decreases the number of required task descriptions. To achieve this, we introduce the concept of *Linked Objects*.

A linked object is best thought of as a virtual object that can be associated with a perceived object within the POM. Linked objects are the entities on which the object-related tasks are formulated and which are constituting to the objects to be considered for collision avoidance.

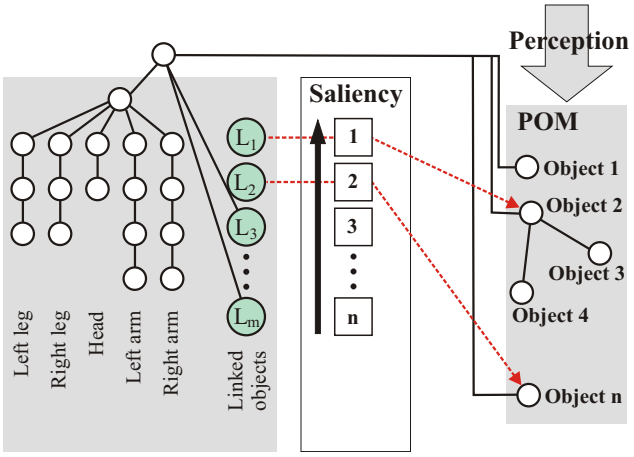

Fig. 3 Linked objects L_i are associated with perceived objects within the *Persistent Object Memory* (POM) using the saliency mechanism. The left part of the tree depicts the kinematic topology of the robot.

The links are created using the attention mechanism explained in the previous section, which provides an ordered list of salient objects. Salient objects of this list are directly assigned to the linked objects in their order: The linked object L_1 refers to the most salient object, linked object L_2 to the second-most salient object etc. If an object's saliency is below a threshold, the association to a linked object is deleted, and the linked object refers to the world reference. This is depicted in Figure 3. Linked object L_1 is associated with object 2 and has the highest saliency. L_2 is associated with object n.

During interaction, one can now use the attention mechanism to indicate the important objects to the system. The tutor can increase an object's saliency by shaking or pointing to it, and decrease the saliency by hiding it. If an object's saliency exceeds another object's saliency, the list will be re-sorted, and the link associations will be updated.

Linked objects allow interesting generalization capabilities for the robot. Formulating a task that relates to linked objects gives the flexibility to interactively change the robot's behavior. For instance, formulating a task to gaze at linked object L_1 leads to a behavior where the robot will always track the most salient object in the scene. The same applies for reaching or approaching an object. If the robot should for instance reach for object 3, the tutor simply points to it before she/he makes the robot perform the reaching movement with a task relating the hand position to the linked object L_1.

In this work, we give the tutor full control over the learning process. Therefore, we limit the saliency computation to the mentioned interaction cues and ignore

more sophisticated features such as color or mass. Because the tutor can restrict the robot's focus, complex scenes with many objects become tractable.

3.6 Attention-based gazing

The saliency mechanism also defines 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 $\boldsymbol{p}_\mathrm{g}$ is calculated according to Eq. (2). The vector $\boldsymbol{p}_i$ corresponds to the position of object i and scalar s_i to its saliency value. Only objects with confidences $conf_i$ that are high enough (larger than constant c_s) are involved in the calculation. If no object is salient, a default gazing direction is activated instead.

$$\boldsymbol{p}_\mathrm{g} = \sum_i \frac{\boldsymbol{p}_i s_i w_i}{\sum_i s_i w_i} \quad \forall i : conf_i > c_\mathrm{s} \tag{2}$$

Additionally, we include a factor w_i 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 by the tutor) objects in its view. This reactive gazing behavior can also be influenced by elements from the top layer of the 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. This gazing behavior is motivated by the selected scenario, and not particularly conform with psychological findings, which revealed that humans cycle their attention among the most salient objects. However, the implemented gazing behavior is a good compromise between keeping objects in the robot's field of view while still enabling it to give feedback. Nevertheless, more biologically plausible attention models will be subject to future work.

Note that this way of including a task-driven mechanism can be regarded as a simplified version of the well-established Theory of Visual Attention (TVA, Bundesen 1990). TVA is capable of explaining a large range of psychological data and it has the hypothesis that attention is guided by a product of a top-down task-dependent so-called pertinence value, here modelled as w_i, and bottom-up object-related so-called attentional weights, here the s_i.

4 Movement learning

A key element of the presented architecture is the ability to learn movements by observing a human tutor. We apply a probabilistic movement representation to exploit the statistical characteristics of a presented movement task in three steps. First, multiple demonstrations of the movement are segmented. Second, a feasible task space is chosen which encodes the movement in a generic way. And finally third, the demonstrations, projected into the chosen task space, are learned using Gaussian Mixture Models.

4.1 Movement segmentation

For the movement segmentation, we exploit the assumption of an interactive scenario with a human tutor to recognize when significant object-related actions are performed.

We propose a movement segmentation that is based on correlative features between the tutor's hand and objects. This is motivated by the chosen experimental setup. However, the concept is also applicable to other correlative features, such as e. g. foot-object relations. The basic idea is that if an object and a hand are close to each other and begin to move coherently, 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. Such a segment is taken as one demonstration of the movement task. This concept works very well for tasks that involve picking and placing objects. However, it should be noted that it is not suited to segment movements that depend on hand gestures, or in which the hand movement differs from the object movement.

For the segmentation a function f is calculated that consists of two terms (Eq. (3)). The function describes the correlation of the hand and object movement.

$$f(\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{v}_1, \boldsymbol{v}_2) = \frac{1}{2} \cdot g(|\boldsymbol{p}_1 - \boldsymbol{p}_2|) + \frac{1}{2} \cdot h(\boldsymbol{v}_1, \boldsymbol{v}_2) \tag{3}$$

The first term g (Eq. (4)) depends on the distance between the two closest points $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ of hand and object, respectively. We compute these distances using swept volume shape approximations of hand and object. Function g uses the Sigmoid function from Eq. (1) to generate values near 1.0 if the hand is near the object.

$$g(d) = \varsigma(c_1 \cdot (d - c_2)) \tag{4}$$

The second term h is similar to g but depends on the velocities of both points.

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

The value of h increases if the difference of the velocity vectors $\boldsymbol{v}_i$ is low. Function h incorporates another term α that depends on the absolute velocities of hand and

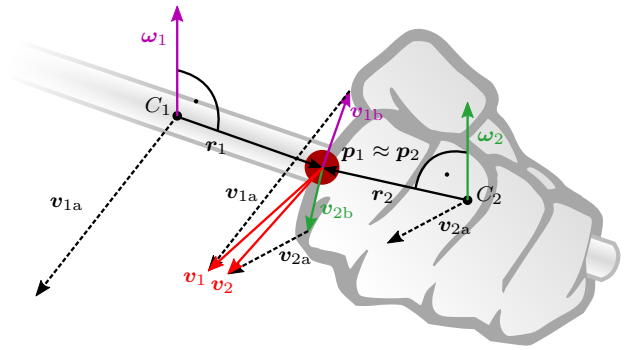

Fig. 4 Calculation of the correlation of velocities of two corresponding objects: Using the closest points makes the result independent of the objects shape.

object. It ensures that function h cannot contribute to the value of f if hand and object are not moving.

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

The constants c_i are used to adjust the value ranges and define how smooth the individual Sigmoid functions transition from 0.0 to 1.0. From the eight constants, those with an odd subscript define the steepness of the transition, while those with an even offset can be seen as a threshold for when this transition occurs.

The values of function f follow a trimodal distribution. If the hand and the object are not moving or are not near each other, then f is a value near 0.0. If hand and object are close to each other, but not or only slightly moving, f is around 0.5. If hand and object are close and their nearest points are moving into similar directions with similar velocities, the value of f increases towards 1.0. This behavior is very advantageous, because it allows to segment object movements by simply applying two thresholds for start and end.

Concerning the velocities used in the calculations, the naïve approach is to use the linear velocities of hand and object. This, however, neglects the objects' shapes and changes of orientations and is a problem as e. g. depicted in Figure 4. When rotating the hand with the stick, their linear velocities $\boldsymbol{v}_{1a}$ and $\boldsymbol{v}_{2a}$ are very different although both are moving together.

To overcome this problem, the angular velocities $\boldsymbol{\omega}_i$ and the nearest points $\boldsymbol{p}_i$ of object and hand are included in the calculation. Computing the radii $\boldsymbol{r}_i$ with a proximity computation between hand and object, the overall velocities $\boldsymbol{v}_i$ of the nearest points can be calculated with:

$$\boldsymbol{v}_i = \boldsymbol{v}_{ia} + \boldsymbol{\omega}_i \times \boldsymbol{r}_i \qquad (7)$$

4.2 Task space selection

After movements have been segmented, a feasible set of control variables needs to be chosen for each set of segments. In Section 5 a task-level control approach is presented that allows to define tasks that relate any body in the kinematic tree representation to any other body. This flexibility however makes it necessary to find a good representation of an observed movement task that is compact, describes the important elements of the task, and does not constrain the robot more than necessary. We therefore include the selection of a feasible task space representation in our learning framework as first proposed in (Mühlig et al. 2009b). The idea is to evaluate commonly used task spaces (*task space pool*) and to select the one in which the movement is represented most consistently. The task space pool includes all relations between the objects that fulfill the following criteria.

The task spaces in the task space pool are predefined. The system makes use of the attention mechanism and the linked objects concept to restrict a possible task description to the most salient objects. Only the first two most salient objects L_1 and L_2 in the scene are regarded in the learning scenario. Dealing with larger combinations of objects will be subject to future work.

The frame of reference in which a movement is described can play a major role for generalization. For each relation of two objects we therefore include the both possible descriptions: The movement of object L_1 in the reference frame of object L_2 and vice versa.

Further, it can be useful to describe object movements not as movements of their center points, but as movements of specific feature points that are also inherent to other objects. For example, if the task is to learn how to place objects on top of another, representing the movement in relative coordinates of the center points of two objects does not generalize over different object sizes. A better way is to describe the movement of one object's bottom wrt. another object's top. Such feature points (top, center, bottom) are included in the object description of known objects and defined a priori. In future this could be combined with mechanisms to detect such features automatically.

We regard the positional and rotational elements of the task space separately, where positions are represented in Cartesian coordinates and orientations as one angle around the normal axis of the camera plane.[1]

For choosing a good task space representation, multiple demonstrations of the same task need to be shown by the tutor. The idea is to compare task spaces and find

[1] The restriction to only one angle results from the use of stereo vision.

the one in which the movement is represented most consistently. This is reflected by a low inter-trial variance of the movement in this particular task space. For this, we calculate a consistency score C for the projection of the object movement into each task space in the task space pool individually and finally choose the task space with the highest score.

The first step is to project all trajectories of the demonstrations into the task space (e. g., the task space relating the bottom of the most salient object to the top of the second-most salient). Then the trajectories are temporally normalized using Dynamic Time Warping (Sakoe 1978; Calinon et al. 2007). This is necessary because the tutor may have presented the task at different speeds. After this normalization the mean trajectory and the inter-trial variance between the demonstrations is calculated. The score for the task space then results from applying the following reciprocal function to the task space trajectory of length N:

$$C = \sum_{t=1}^{N} \frac{1}{1 + s \cdot \sigma_t} \tag{8}$$

The scalar σ_t is the inter-trial variance of the demonstrations at timestep t and s is a scaling parameter. We chose a fixed value for s, but it can also be used to scale the variance for each task space individually. The score can be interpreted as the consistency of the task space for this specific task. It reaches high values only if there are phases of low inter-trial variance.

The suitability of each task space to represent the movement is now characterized by this value and the task space with the highest score is selected.

4.3 Probabilistic movement representation

After a feasible task space has been chosen and the task demonstrations are temporally normalized using Dynamic Time Warping, a representation is learned by means of Gaussian Mixture Models (see also Calinon 2009):

$$p(\boldsymbol{x}_i) = \sum_{k=1}^{K} \pi_k p(\boldsymbol{x}_i|k) \quad \text{with} \quad p(\boldsymbol{x}_i|k) = \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{9}$$

The vectors $\boldsymbol{x}_i$ correspond to the time series of all demonstrations, defined in the chosen task space with time as an additional dimension. The symbols π_k, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ represent the prior, mean vector and covariance matrix of the Gaussian distributions $k = 1..K$.

This GMM is learned for each primitive, which means that the overall sequence is composed of primitives that have different GMMs and act on different sets of control variables. The sequential flow of the primitives is however not affected by this mechanism. The probabilistic representation allows to store the mean movement and the covariance information that can later be exploited by the robot. The idea is that a high inter-trial variance corresponds to a less important phase of the demonstrated task. With the movement optimization scheme presented in the next section, we allow the robot to diverge from such less important phases of the trajectory to fulfill additional criteria.

The Gaussian Mixture Model is learned with a batch learning approach using Expectation-Maximization and k-Means initialization. The optimal number of Gaussian distributions K used for this representation is dependent on the complexity of the movement and can be estimated using the Bayesian Information Criterion (Schwarz 1978). This criterion basically rates the model complexity versus the representation quality. The system applies an efficient heuristic using this criterion, which does not require to perform a full EM learning for each possible number of Gaussians (for details see Mühlig et al. 2009a).

After learning, the movement is stored in form of a movement primitive in the *Movement Primitive Memory*. The learned task is not associated to specific objects, but stored as a relation of linked objects in the task space chosen by the task space selection.

5 Movement reproduction

The presented framework uses movement primitives as basic behavioral building blocks. These movement primitives can be either predefined or learned using the techniques described in the previous section. This section describes how movement primitives are combined to perform more complex tasks, the optimization-based adaptation of movement primitives to new situations and the advantages of decoupling the movement task from the embodiment.

5.1 Sequencing of movement primitives

Most of the real world problems are to complex to be solved with single movement primitives. To overcome this limitation the system includes a hierarchical state chart on the top level of the framework. In this state chart, movement primitives that are either learned or predefined are connected using different kind of transitions. These transitions switch the activation of one movement primitive to the next if certain conditions are fulfilled. For example, such conditions can be the

reaching of the end state of a movement primitive or a specific posture of the tutor (e. g., raising a hand). Usually, movement sequences are defined in the procedural memory which are later augmented with newly learned skills. Which of the sequences should be used with a new skill is decided by the interaction with the human tutor.

As for the experiments, we have manually designed the flow of segments, so that the neighboring segments have a meaningful transition of the selected task spaces. In general, it is not fatal to remove or add a primitive. The system is robust in this sense, since each primitive is implemented by a movement converging to a vector of subsequent attractors. The alignment is implicitly done by the transient behavior of the attractor dynamics. However, it is sometimes necessary to check that certain preconditions are met. For instance, when grasping an object, we must take care that the hand has been opened before. Or before bringing the robot in its resting pose, it needs to be made sure that it avoids collisions with the table, either by using optimization, or by designing a safe sequence.

It needs to be noted that the sequencing layer is not in the focus of the current system and is rather an enabling functional module. The sequencing layer primarily serves to enable the execution of the learned skills with new objects or in different ways. Recently, much more sophisticated methods of planning with movement primitives have become available (e. g., Toussaint et al. 2010) and may inspire future extensions of our system.

5.2 Movement optimization

For movement reproduction, the movement primitive layer applies a linear attractor system to the selected task descriptions. This leads to a smooth movement that converges the robot's pose to the attractor target values. We developed movement primitives with different levels of complexity. Simple movement primitives converge the robot's trajectory reactively, with mechanisms to locally avoid joint limits or collisions. These primitives are used to realize the preparatory movements like e. g. reaching and grasping an object. This is however not enough to adapt complex, learned movements to new situations. For instance if a stacking task has been learned on an empty table, and is to be imitated in presence of an obstacle, the movement optimization modifies the movement to avoid the collisions with the obstacle. Further, criteria like actuator limits or collisions with obstacles need to be incorporated. To account for this, we developed primitives that are composed of several attractor vectors that can be optimized with respect to a set of criteria, such as collision and joint-limit avoidance (for details see Toussaint et al. 2007). The employed optimization scheme works in an optimal control fashion and uses a modified version of the R-Prop algorithm (Igel and Hüsken 2003). It is local in space, but anticipates a future time horizon. The input is the current state of the system (robot and object poses), the probabilistic model of the learned movement (GMM), as well as a set of weighted cost function terms. The chosen cost function comprises terms to avoid collisions, joint limits, and to generate a smooth movement. With this input, the system computes an optimal sequence of attractor vectors that governs the overall movement. In decently complex environments, this scheme allows to generate movements that are collision-free and optimal in other respects. Imitation information is included consistently as a cost function that describes the similarity of a learned movement $\hat{\boldsymbol{\mu}}_t$ to the movement of the robot $\boldsymbol{x}_t$:

$$c_{\text{im}} = \sum_{t=1}^{T} (\boldsymbol{x}_t - \hat{\boldsymbol{\mu}}_t)^T \boldsymbol{W}_t (\boldsymbol{x}_t - \hat{\boldsymbol{\mu}}_t) \tag{10}$$

The similarity is weighted with $\boldsymbol{W}_t$, which depends reciprocally on the covariance at the corresponding time point. Both, information about the mean and covariance is extracted from the respective movement primitive by applying Gaussian Mixture Regression (GMR) to the previously learned Gaussian Mixture Model.

Applying this optimization scheme results in imitated movements that reflect the tutor's characteristics precisely in phases with low inter-trial variance, while phases with higher variance weight the other criteria stronger. This means the system imitates the movement as good as possible, but does adapt the movement to account for its constraints and limitations.

In order to assure a fluent interaction we implemented the possibility of parallel optimization. Using the information of the state chart about predicted future states, it is possible to optimize movements in advance. Assuming a static environment without the need of a re-optimization when the optimized movement primitive is reached, this leads to a fluent transition between the movement primitives.

5.3 Movement control and task coordinates

The robot's kinematics is described in the form of a tree structure depicted in Figure 5. The individual links are connected by degrees of freedom (joints) or rigid body transformations. The tree also comprises links which represent objects from the environment and are updated by the POM. This allows to derive the kinematics not only with respect to a heel or world reference frame, but also to formulate robot-object or object-object relations.

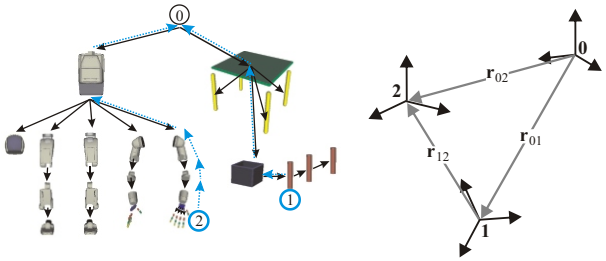

Fig. 5 Left: Kinematic tree, both robot and other scene elements are represented in one consistent graph. - Right: Coordinate systems to compute relative body kinematics.

Kinematic task descriptor are defined as the relative movement of one link with respect to any other link. To mathematically formalize this concept, we look at the relative kinematics of an articulated chain, such as depicted in Figure 5 right. Coordinate frame 0 denotes the root. Frame 1 is an arbitrary body which is connected to 0 through a set of joints. Body 2 shall be represented relative to body 1 with vector $\boldsymbol{r}_{12}$. The kinematic equations can now be written as follows:

$$\boldsymbol{r}_{12} = \boldsymbol{r}_{02} - \boldsymbol{r}_{01} \qquad \dot{\boldsymbol{r}}_{12} = \dot{\boldsymbol{r}}_{02} - (\dot{\boldsymbol{r}}_{01} + \boldsymbol{\omega}_1 \times \boldsymbol{r}_{12}) \quad (11)$$

The outer product term of Eq. (11) right is due to the angular velocity $\boldsymbol{\omega}_1$ of body 1. Introducing the coordinate system in which the respective vector is represented as the left sub-index and projecting the velocities into the robot's configuration space with the respective translational $\dot{\boldsymbol{r}}_i = \boldsymbol{J}_{\mathrm{T},i}\ \dot{\boldsymbol{q}}$ and rotational Jacobians $\boldsymbol{\omega}_i = \boldsymbol{J}_{\mathrm{R},i}\ \dot{\boldsymbol{q}}$, the differential kinematics gets

$$ {}_1\dot{\boldsymbol{r}}_{12} = \boldsymbol{A}_{10}\left({}_0\boldsymbol{J}_{\mathrm{T},2} - {}_0\boldsymbol{J}_{\mathrm{T},1} + {}_0\tilde{\boldsymbol{r}}_{12}\ {}_0\boldsymbol{J}_{\mathrm{R},1}\right)\dot{\boldsymbol{q}} = {}_1\boldsymbol{J}_{\mathrm{T,rel}}\ \dot{\boldsymbol{q}} \quad (12)$$

with $\tilde{\boldsymbol{r}} = (\boldsymbol{r}\times)$ being a skew-symmetric matrix representing the outer product, and $\boldsymbol{A}_{10}$ being a rotation matrix from frame 0 to frame 1. The task descriptors for a segment's spatial orientation can be computed accordingly, for instance in Euler (3D) or Spherical angles (2D), or as the inclination of one body axis with respect to any other (1D). With these equations, one can formulate task descriptors that relate any link of the tree to any other. Further, it is possible to compute these descriptors element-wise, such as "position of body 2 with respect to body 1 in X-direction", or "Euler α angle of body 2 with respect to body 0".

Similarly, we can derive dynamic task descriptors for the overall linear and angular momentum and others, see Gienger et al. (2010b) for details.

For a set of task descriptors, we augment an overall task Jacobian, and compute the joint rates with an inverse kinematics scheme based on the concept presented in Liégeois (1977)

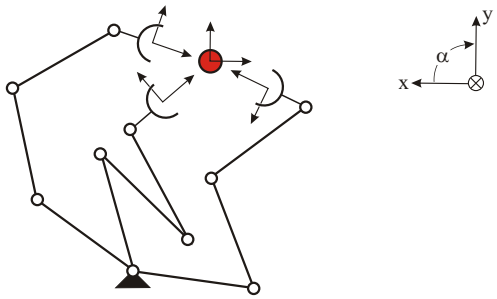

Fig. 6 Different robot postures according to the same task represented in effector coordinates: All three depicted poses correspond to the same coordinates $(x\ y\ \alpha)^T$.

$$\delta\boldsymbol{q} = \boldsymbol{J}^{\#}\ \Delta\boldsymbol{e} - \alpha\ (\boldsymbol{I} - \boldsymbol{J}^{\#}\boldsymbol{J})\boldsymbol{W}^{-1}\left(\frac{\partial H}{\partial \boldsymbol{q}}\right)^T \quad (13)$$

where $\boldsymbol{J}^{\#}$ is a $\boldsymbol{W}$-weighted Pseudo-Inverse of the augmented task Jacobian, $\Delta\boldsymbol{e}$ is the feedback error of the task coordinates, and H is a secondary objective (we chose a joint limit avoidance penalty) whose gradient is projected into the null space of the movement through the right term of Eq. (13).

The choice of the order of the relative coordinates yields some interesting aspects. This is illustrated in Figure 6 for a simple planar redundant system described by task variables $(x\ \ y\ \ \alpha)$. If the task variables are represented in the object's or robot's frame of reference, different values are needed to realize the depicted poses. If, like depicted, the orientation between object and end effector is not important, it may be more advantageous to represent the task variables in the effector's frame of reference. In that case, all three poses can be realized with the same values $(x = d,\ y = 0,\ \alpha = 0)$. This task description introduces an invariance with respect to the relative pose between effector and object. Its null space comprises the relative pose between effector and object. When resolving redundancies with Eq. (13), the achieved pose will correspond to a (local) optimum with regard to the cost function H. This difference is exploited when selecting a feasible task space as explained in Section 4.2.

5.4 Body schema adaptation

Kinematic structures as depicted in Figure 5 represent a parent-child hierarchy: The movement of a segment will affect the movement of its children. In many practical situations, changes to this kinematic configuration occur. An example is a robot grasping an object and putting it at a different position. Another example is to put an object from a table on a tray which is placed on the table. A common approach to deal with such changes is

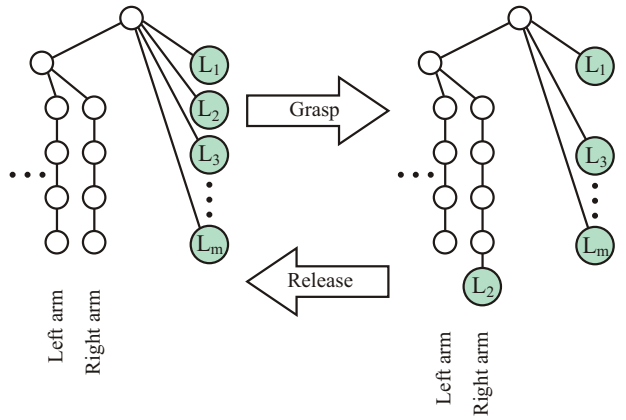

Fig. 7 Adaptation of the kinematic chain according to performed action. The linked objects are denoted with L_i, index i being the saliency index.

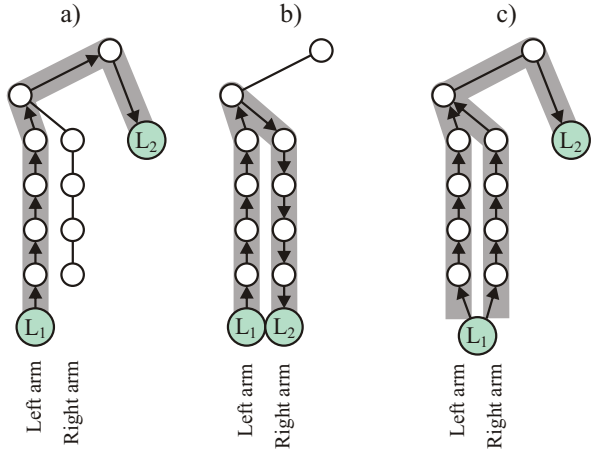

Fig. 8 Kinematic chains for different body schemas. The gray lines cover the joints and transformations that are involved in the movements.

to keep the kinematic configuration, but to compute the robot's end effector coordinates based on the desired object transformation. This way, the movement can be controlled in end effector coordinates, and collisions can be taken into account by applying avoidance strategies based on the transformed object geometry.

We propose to address this problem by adapting the body schema (or body image), which commonly refers to the perception of a human's physical appearance. In robotics, a number of approaches to learn and adapt the body schema have been proposed, for instance for proprioceptive models (Hersch et al. 2008) and for models including tools (Stoytchev 2003; Nabeshima et al. 2006).

In the following, we assume the geometric properties of the system to be known, and focus on dealing with structural changes during interaction with the environment. We argue that kinematic structure modifications can be modelled in a higher abstraction of the movement generation system, such as in actions or in action sequences. For instance if a robot "grasps" an object, it is either known or it can be reconfirmed by sensor feedback that the grasp is successful and the object is held by the robot's end effector.

We suggest to exploit this knowledge and apply such structural modifications based on actions like grasping or releasing an object. This is depicted in Figure 7. Applying an action that grasps linked object L_2 will modify its connectivity so that it is connected to the grasping hand of the system. It should be noted that this also accounts for the case where the linked object refers to a parent-child structure like object 2 in Figure 3. An example would be to grasp a tray on which two objects are placed. In the same way, releasing the linked object can be associated with connecting it to the world's frame of reference, or any other object at which it is positioned.

This approach is beneficial, since firstly, an abstraction of the embodiment is introduced. Object movements are generic, while the movement of an end effector always incorporates the knowledge about a specific transformation between end effector and object. Secondly, representing movements in object coordinates allows to introduce invariance in the same line of argument as discussed in Figure 6: Stacking a cylinder on top of another can be described by aligning the cylinders symmetry axis, while it is rather difficult to find a general end effector-object relation.

Figure 8 illustrates this for three examples. Let's assume a task descriptor that relates the transformation of linked object L_1 to the transformation of linked object L_2. The target values are determined to put L_1 on top of L_2. In example a), L_1 is connected to the left hand, while L_2 has a fixed transformation. The system will generate a trajectory moving the grasped L_1 on L_2 with its left arm. In case b), both L_1 and L_2 have been grasped. The result is a coordinated bimanual movement, L_1 is put on L_2 which is held with the right hand. In case c), L_2 has again a fixed transformation in world coordinates, and L_1 has been grasped with both hands. In this case, the system will put L_1 on top of the static object L_2, but this time generating a coordinated bimanual trajectory with the grasped object L_1.

6 Experiments

In this section we present two experiments to illustrate the generalization concepts of the framework. The supplementary video (Online Resource 1) shows the two experiments in different situations. It includes also an

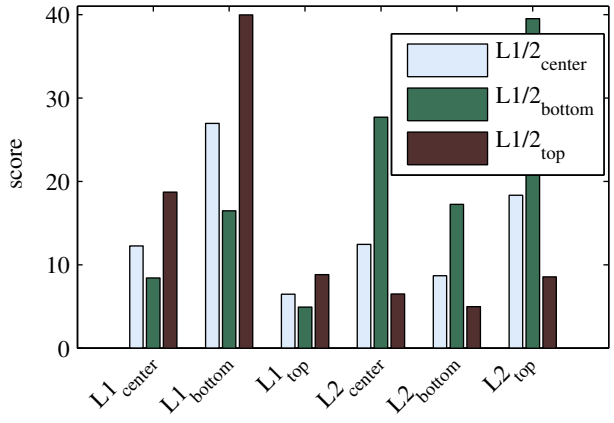

Fig. 9 Consistency scores for the stacking experiment. The task is represented best by relating the bottom of the most salient object ($L1_{bottom}$) to the top of the the second-most salient object ($L2_{top}$). The feature points "bottom", "center" and "top" have been predefined to match the object geometry.

example where the robot avoids collisions while imitating a movement, which is not addressed in this paper (but see Mühlig et al. 2010).

6.1 Stacking objects

In the first experiment the robot is taught how to put objects on top of another. Figure 10 shows the setup and the interaction flow between tutor and robot. In the beginning the tutor taps on the red object to make it salient to the robot. Internally, this associates the red object to linked object L_1 and the green object to L_2 since only two objects are present. Then, the tutor puts the red object onto the green object. The relevant movement is segmented by evaluating the correlation of hand and object movement as described in Section 4.1. After the demonstration the robot gazes at the tutor and waits for a response. The tutor raises his left hand to confirm that the movement shall be learned.

For brevity, only one demonstration of the task is depicted in the figure. The tutor however demonstrated the task multiple times (between three and seven demonstrations have been found to be reasonable for the selected scenario) and with varying objects (see Figure 11 top left). This allowed the automatic task space selection to choose the most reasonable task space, namely relating the bottom of the most salient object L_1 to the top of the second-most salient object L_2.

Next, the tutor raises both hands to instruct the robot to perform the task bimanually. This starts several processes in parallel. Internally, the system first learns the movement using the probabilistic encoding explained in Section 4. Then the optimization process is initialized with the predicted pose of the robot. Thereafter the solution is computed while the robot approaches the table. When the robot stands in front of the table, it grasps the objects, performs the learned task using the optimization result, puts the objects back on the table and returns.

Note, that the learned movement was integrated into the movement sequence for grasping both objects. If the tutor would have raised only one hand instead of both hands, the same learned movement primitive would have been integrated into another movement sequence in which the robot would have performed the task one-handed. We want to emphasize that the preparatory movements like approaching the table or grasping the object have been pre-programmed and are not learned. Imitating the task one-handed or bimanual is encoded as separate sequences within the state chart. For the one-handed case, the body schema is modified such that the most salient (red) object is topologically assigned to the left hand (Figure 8 a), while the second-most salient (green) object has a fixed transformation to the world reference frame. For the bimanual case, both objects are attached to the hands (Figure 8 b).

Figure 9 depicts a comparison of the calculated task space scores. According to Section 4.2, 18 different task spaces are compared. These are the positions of the feature points of L_1 and L_2, projected into all possible task spaces. The orientation has been excluded from the task space selection. The second-best score corresponds to the same feature relation as the best score, with the difference that the movement is represented in the other object's reference frame. This results from the nature of the task, which makes both representations equally effective. This however is not always the case as explained in Section 5.3.

For emphasizing the quantitative influence of choosing a good task space, Figure 12 compares the positional task elements of a typical representation relating object centers with the chosen task space representation. The bottom row shows the score value for each time step.[2] It can be seen that the chosen representation is less variant in the end of the movement, especially in Z-direction (vertical direction).

Representing the task in these coordinates allows the robot to apply it to unknown objects, if they comprise information about their top and bottom. This generalization ability is demonstrated by instructing the robot to put a candle on a candle holder (see Figure 11 bottom left). The movement was not learned with these two objects. Further, the top of the candle holder is not above its center, as compared to the cylindrical object

[2] The term within the sum of Eq. (8) plotted for each time step.

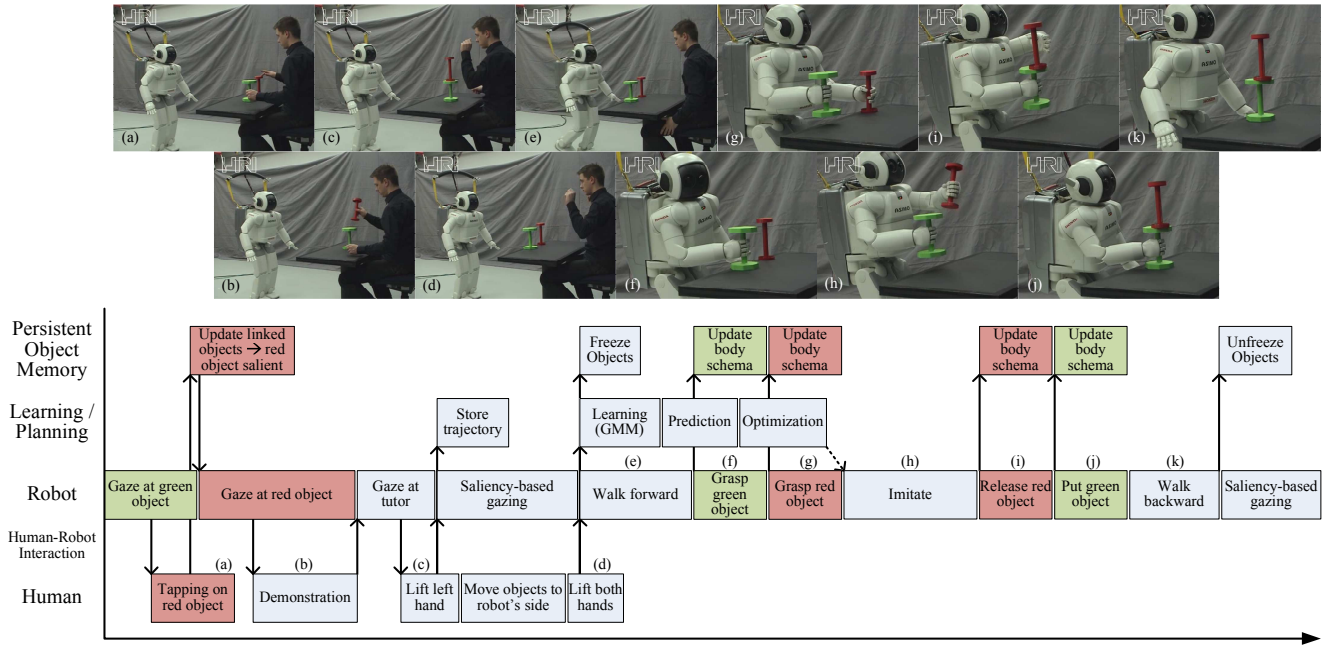

Fig. 10 Illustration of the interaction and the interplay between internal elements during an experiment. Note that the movement optimization is done in parallel to the robot's movement. The dashed line indicates that the imitation phase may have to wait for the optimization to finish. During phases between "freeze objects" and "unfreeze objects", the scene is not updated by the vision system and assumed to be static.

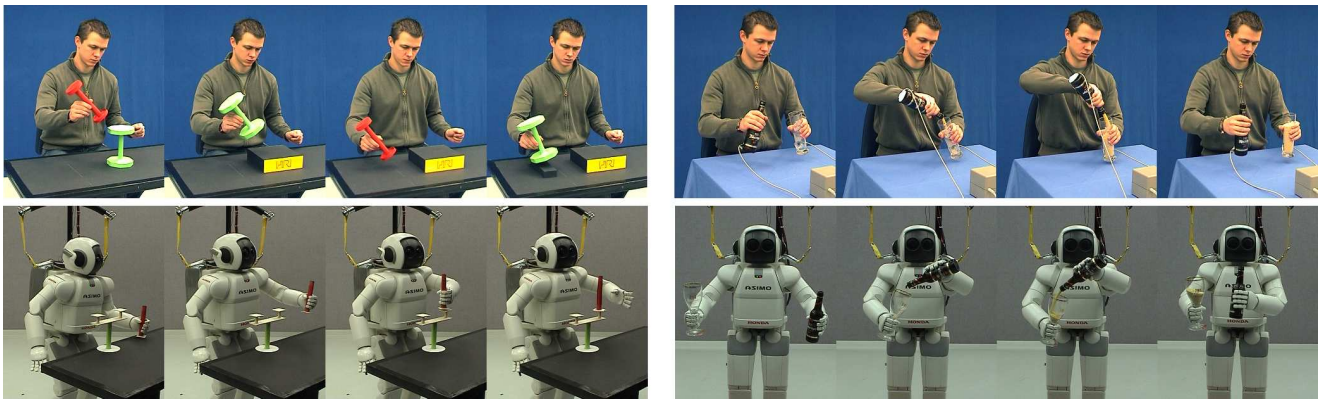

Fig. 11 Two experiments were conducted. Left: The tutor is stacking various objects on top of another and the robot is able to reproduce the movement with two objects not seen before. Right: The tutor is demonstrating how to pour a beverage and the robot is successfully able to reproduce the movement.

the movement has been learned with. If the task would have been learned in the coordinates of the object center, the robot would not have been able to reproduce it correctly.

Additionally to the selection of a feasible task space, the experiment shows how the body schema adaptation can be used for generalization. In the experiment, the tutor demonstrated only one-handed movements with the target object remaining on the table. The robot however was able to perform the movement bimanually (see Figure 10). This results from the movement being represented in relative object coordinates and the body schema adaptation. When grasping an object, it becomes attached to the kinematic tree of the robot, such as described in Section 5.4. The optimization process then adapts the robot's motion to comply with the desired object motion. Note that the movement primitive for this bimanual task included an additional constraint for holding the green object upright.

6.2 Pouring

In the second experiment the robot had to learn how to pour a beverage from a bottle into a glass (Figure 11 right). The objects were tracked with a magnetic-field-based motion tracking system, the tutor was still

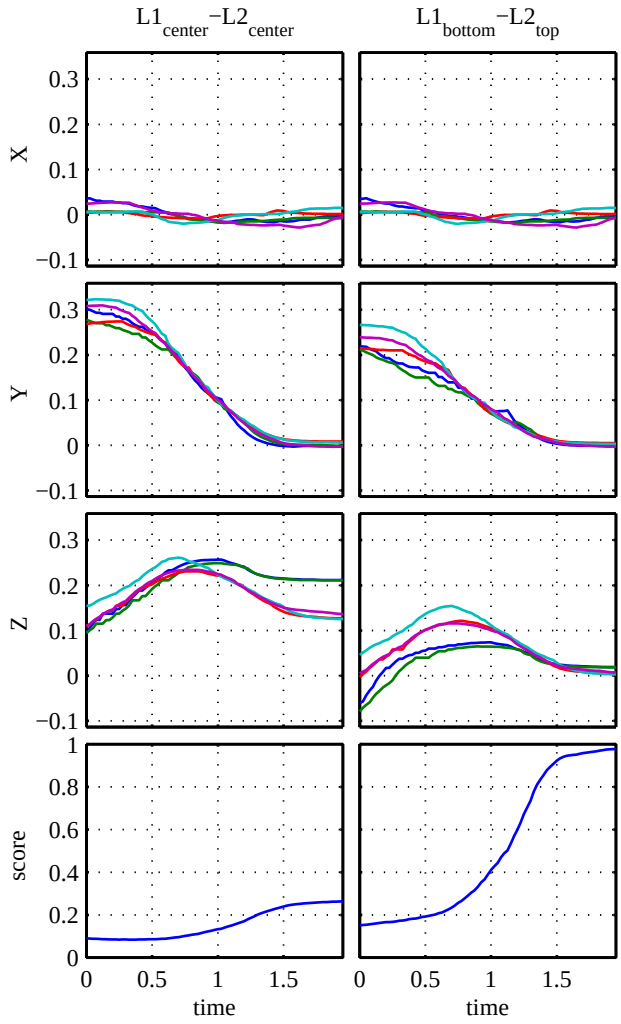

Fig. 12 Comparison of the five stacking demonstrations in the naïve representation relating object centers (left) with the chosen representation (right). Plots with the same line color belong to the same demonstration.

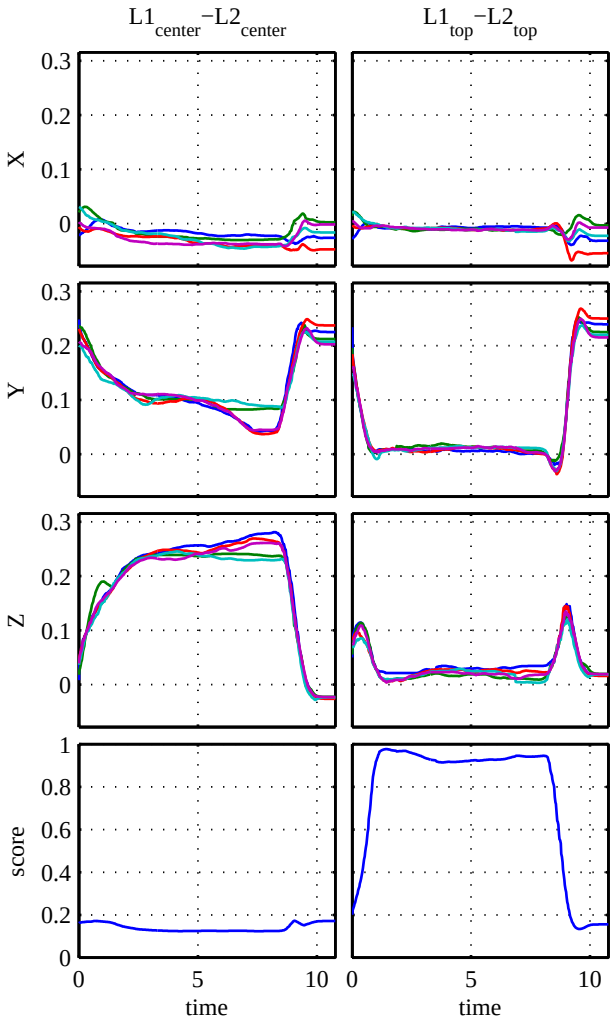

Fig. 13 Comparison of the five pouring demonstrations in the naïve representation relating object centers (left) with the chosen representation (right).

detected through vision. Further, we used millet as a replacement for liquid.

After the tutor presented the task five times, the system automatically chose the most appropriate task space. As can be seen in Figure 14, the best representations for the task is to relate the top of the bottle to the top of the glass or vice versa. Again we show a comparison between the naïve representation relating object centers and the chosen task space in Figure 13. It can be seen that the chosen representation yields a lower variance especially during the important middle phase (the actual pouring phase).

Note, that due to the task space selection, the top of the bottle wrt. the top of the glass is always tracked precisely. Even in difficult situations where the orientation of both objects cannot be reproduced as observed,

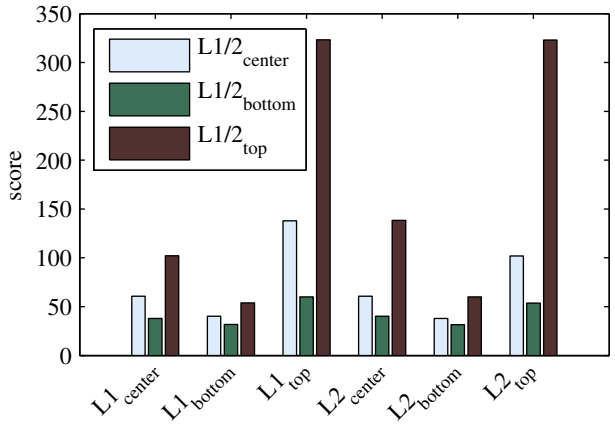

Fig. 14 Consistency scores for the pouring experiment. The task is represented best by relating the top of the most salient object ($L1_{top}$) to the top of the the second-most salient object ($L2_{top}$) or vice versa.

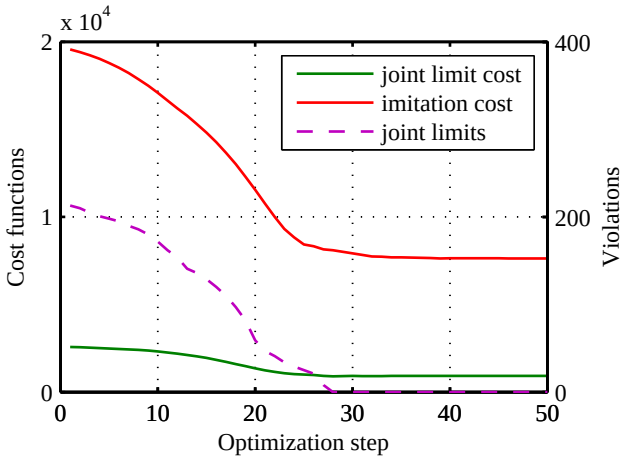

Fig. 15 At the beginning of the optimization the learned movement is not tracked precisely (high imitation costs) and violates joint limits. After 50 iterations a valid movement emerges.

the robot is still able to pour correctly. This would not be the case with a representation that uses the object's center points.

Figure 15 also shows the necessity of the optimization process (see Section 5.2) to adapt the movement to the robot's embodiment. The plot depicts the two most influencing cost terms and the number of violated joint limits. Without optimization, 213 joint limits were violated.[3] After 50 iterations, the optimization generated a valid movement that tracks the learned movement as good as possible. For the movement imitation, bottle and glas have been properly given to the hands in order to match the reference points coming from the observations.

7 Discussion and outlook

The presented system addresses the problem of object movement learning from a whole systems perspective and thereby provides a more comprehensive strategy than other, more restricted systems. The main goal is to demonstrate generalization of these learned skills to new objects and to new ways of execution, e. g. with a different or both hands. For achieving this goal, a large number of functional modules for visual perception, action generation, movement control, movement representation and attention have been combined and integrated. We are aware that for some of these components there are more sophisticated and possibly more powerful methods available, if considered in isolation. But when building and describing a complete system, we had to use shortcuts in certain functional modules and use heuristics in favor of the overarching goal to reach the desired learning and generalization behavior. However, there are important novel concepts in our system, which in combination provide a leap forward towards a more flexible learning and execution of learned skill. We elaborate more on these mechanisms in the following.

From a cognitive point of view, each learning system must comprise at least three major elements, regardless of what kind of representation or learning is used. First, an abstraction process has to decontextualize the invariants to be learned from particular example data, which includes to select and acquire such data. Second, a representation of this invariant has to be stored for later reuse, and third, adaptive processes to readjust the learned representation towards new situations are needed for true generalization. Much of the effort and novel ideas we have presented, serves to realize these three steps in tight interaction with a user and we discuss the three aspects in turn.

Many movement learning systems assume that example trajectories are properly segmented and already recorded in the appropriate space (e. g., Calinon and Billard 2008; Kormushev et al. 2010; Pastor et al. 2009; Ijspeert et al. 2003). If user interaction to record data is provided, most often explicit segmentation by the user is required (e. g., Nicolescu and Matarić 2003), i. e. the interaction follows the special requirements of the robot learning system. In the attempt to release these strong restrictions, we provide the system with the segmentation mechanism describe in Section 4.1. It turns out that the articulated scene model, i. e. an internal representation of the user and the relevant objects is important to achieve the necessary robustness in this process, because filtering for sensory noise, occlusions etc. need to be resolved. On the representational side, we use a probabilistic encoding of the example trajectories in a Gaussian Mixture Model as several authors before (Calinon 2009). The novel concept of linked objects in connection with the encoding in relative coordinates however allows for a much more flexible reuse, because the stored invariant is the shape of the trajectory that is given with respect to a kind of placeholder for concrete objects - the linked object. The system further has the ability to choose the task space for encoding autonomously, which yields great generalization possibilities often neglected by other systems. Finally, one of the biggest strengths of our system is to provide very flexible means for re-adaptation of the learned representations to a new situation. To be able to perform a task with different objects, a different arm, or bimanually instead of single-handed requires the combination of several ideas. The key novel concept is our flexible

[3] The number of joint limit violations is cumulative over all controller steps of the movement and can therefore be higher than the actual number of joints.

scheme for reconfiguration of the body kinematics (Section 5.4) that allows to resort to existing path planning and control methods even when one or more objects are moved. To this end, the earlier introduced flexible attractor based movement generation scheme provides the required adaptivity to adjust the movement to changed collision constraints and, as was already demonstrated in our previous work Mühlig et al. (2010), to obstacles. And finally the attentive mechanism that is based on the internal model allows to interactively determine the concrete objects to manipulate.

Of course our system also has limitations, especially on the perception side and in the movement sequencing layer. First, we don't track the visually perceived objects while the robot imitates the movement. This is due to the problems imposed by occlusions and other inaccuracies, and severely limits the system in dynamically changing environments. Second, the 3D object shapes and the feature point locations (top, center, bottom) have been predefined for the experiments. This could be improved by vision-based approaches. Another limitation is related to the observable sensory modalities: The system can only learn the relation of kinematic data, but can't learn the force and torque interaction between objects. And lastly, a more continuous and parallel activation of movement primitives needs to be investigated. Although the currently applied state chart approach eases the modelling of movement sequences, it creates other challenges such as handling failures during task execution in dynamic environments.

Despite these limitations, our system goes in its complete realization beyond previous work in whole systems robot learning of movement skills, because it has previously not been demonstrated how a learned skill can be executed with various object configurations and differently from the demonstrations with both hands.

Acknowledgements The authors thank Dr. Tobias Luksch for his help during conducting the experiments.

References

Acosta-Calderon CA, Hu H (2005) Robot imitation: Body schema and body percept. Applied Bionics and Biomechanics 2:131–148

Asfour T, Gyarfas F, Azad P, Dillmann R (2006) Imitation Learning of Dual-Arm Manipulation Tasks in Humanoid Robots. In: International Conference on Humanoid Robots, 2006 6th IEEE-RAS, pp 40–47

Azad P, Asfour T, Dillmann R (2007) Toward an Unified Representation for Imitation of Human Motion on Humanoids. In: Proceedings 2007 IEEE International Conference on Robotics and Automation, IEEE, April, pp 2558–2563

Beetz M, Jain D, Mösenlechner L, Tenorth M (2010) Towards Performing Everyday Manipulation Activities. Robotics and Autonomous Systems 58(9):1085–1095

Bohg J, Barck-Holst C, Huebner K, Ralph M, Rasolzadeh B, Song D, Kragic D (2009) Towards Grasp-Oriented Visual Perception for Humanoid Robots. International Journal of Humanoid Robotics 6(3):387–434

Bolder B, Dunn M, Gienger M, Janssen H, Sugiura H, Goerick C (2007) Visually Guided Whole Body Interaction. In: IEEE International Conference on Robotics and Automation (ICRA)

Bundesen C (1990) A Theory of Visual Attention. Psychological review 97(4):523–547

Burghart C, Mikut R, Stiefelhagen R, Asfour T, Holzapfel H, Steinhaus P, Dillmann R (2005) A Cognitive Architecture for a Humanoid Robot: A First Approach. In: Proceedings of 2005 5th IEEE-RAS International Conference on Humanoid Robots, pp 357–362

Calinon S (2009) Robot Programming by Demonstration: A Probabilistic Approach. EPFL Press

Calinon S, Billard A (2008) 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

Calinon S, Guenter F, Billard AG (2007) On Learning, Representing, and Generalizing a Task in a Humanoid robot. IEEE transactions on systems, man, and cybernetics Part B, Cybernetics: a publication of the IEEE Systems, Man, and Cybernetics Society 37(2):286–98

Einecke N, Mühlig M, Schmüdderich J, Gienger M (2011) "Bring it to me" - Generation of Behavior-Relevant Scene Elements for Interactive Robot Scenarios. In: 2011 IEEE International Conference on Robotics and Automation (ICRA)

Eppner C, Sturm J, Bennewitz M, Stachniss C, Burgard W (2009) Imitation Learning with Generalized Task Descriptions. In: Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA), Kobe, Japan

Gienger M, Janssen H, Goerick C (2005) Task-oriented whole body motion for humanoid robots. In: 2005 5th IEEE-RAS International Conference on Humanoid Robots, pp 238– 244

Gienger M, Mühlig M, Steil JJ (2010a) 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 (IROS 2010)

Gienger M, Toussaint M, Christian G (2010b) Motion Planning for Humanoid Robots. In: Harada K, Yoshida E, Yokoi K (eds) Whole-body Motion Planning - Building Blocks for Intelligent Systems, Springer, chap 3

Hecht F, Azad P, Dillmann R (2009) Markerless human motion tracking with a flexible model and appearance learning. In: IEEE International Conference on Robotics and Automation (ICRA)

Heracles M, Bolder B, Goerick C (2009) Fast Detection of Arbitrary Planar Surfaces from Unreliable 3D Data. In: International Conference on Intelligent Robots and Systems (IROS), IEEE/RSJ

Hersch M, Sauser E, Billard A (2008) Online learning of the body schema. International Journal of Humanoid Robotics 5(2):161–181

Igel C, Hüsken M (2003) Empirical evaluation of the improved Rprop learning algorithm. Neurocomputing 50(C):105–123

Ijspeert A, Nakanishi J, Schaal S (2003) Learning Attractor Landscapes for Learning Motor Primitives. Advances in Neural Information Processing Systems pp 1547–1554

Inamura T, Toshima I, Tanie H, Nakamura Y (2004) Embodied Symbol Emergence Based on Mimesis Theory. The Interna-

tional Journal of Robotics Research 23:363–377
Itti L, Koch C, Niebur E (1998) A Model of Saliency-Based Visual Attention for Rapid Scene Analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(11):1254–1259
Khansari-Zadeh S, Billard AG (2010) Imitation learning of Globally Stable Non-Linear Point-to-Point Robot Motions using Nonlinear Programming. In: Proceeding of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 2676–2683
Kober J, Peters J (2009) Learning Motor Primitives for Robotics. In: 2009 IEEE International Conference on Robotics and Automation, IEEE, pp 2112–2118
Kormushev P, Calinon S, Caldwell DG (2010) Robot Motor Skill Coordination with EM-based Reinforcement Learning. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp 3232–3237
Liégeois A (1977) Automatic Supervisory Control of Configuration and Behavior of Multibody Mechanisms. IEEE Transactions on Systems, Man and Cybernetics 7(12):861–871
Lopes M, Santos-Victor J (2005) Visual Learning by Imitation With Motor Representations. IEEE Transactions on Systems, Man, and Cybernetics, Part B 35(3):438–449
Lopes M, Melo FS, Montesano L (2007) Affordance-Based Imitation Learning in Robots. In: Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems
Matarić MJ, Pomplun M (1998) Fixation Behavior in Observation and Imitation of Human Movement. Cognitive Brain Research 7:191–202
McGuire P, Fritsch J, Steil JJ, Röthling F, Fink GA, Wachsmuth S, Sagerer G, Ritter H (2002) 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, pp 1082–1088
Mühlig M, Gienger M, Hellbach S, Steil JJ, Goerick C (2009a) Task-level Imitation Learning using Variance-based Movement Optimization. In: IEEE International Conference on Robotics and Automation (ICRA)
Mühlig M, Gienger M, Steil JJ, Goerick C (2009b) Automatic Selection of Task Spaces for Imitation Learning. In: 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems
Mühlig M, Gienger M, Steil JJ (2010) Human-Robot Interaction for Learning and Adaptation of Object Movements. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010)
Nabeshima C, Kuniyoshi Y, Lungarella M (2006) Adaptive Body Schema for Robotic Tool-Use. Advanced Robotics 20(10):1105–1126
Nagai Y, Muhl C, Rohlfing KJ (2008) Toward Designing a Robot that Learns Actions from Parental Demonstrations. In: 2008 IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, pp 3545–3550
Nakamura Y (1991) Advanced Robotics: Redundancy and Optimization. Addison-Wesley Publishing Company
Nicolescu M, Matarić MJ (2003) Natural Methods for Robot Task Learning: Instructive Demonstrations, Generalization and Practice. In: Proceedings of the Second International Joint Conference on Autonomous Agents and Multi-Agent Systems, ACM, pp 241–248
Nicolescu M, Matarić MJ (2006) 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
Orabona F, Metta G, Sandini G (2007) A Proto-object Based Visual Attention Model. Attention in Cognitive Systems Theories and Systems from an Interdisciplinary Viewpoint pp 198–215
Pastor P, Hoffmann H, Asfour T, Schaal S (2009) Learning and Generalization of Motor Skills by Learning from Demonstration. In: IEEE International Conference on Robotics and Automation
Sakoe H (1978) Dynamic programming algorithm optimization for spoken word recognition. IEEE Transactions on Acoustics, Speech, and Signal Processing 26:43–49
Schaal S, Ijspeert A, Billard AG (2003) Computational Approaches to Motor Learning by Imitation. Philosophical transactions of the Royal Society of London Series B, Biological sciences 358(1431):537–47
Schmuedderich J (2010) Multimodal Learning of Grounded Concepts in Embodied Systems. Berichte aus der Robotik, Shaker Verlag GmbH, Germany
Scholl BJ (2001) Objects and attention: the state of the art. Cognition 80(1–2):1–46
Schwarz G (1978) Estimating the Dimension of a Model. Annals of Statistics 6(2):461–464
Steil JJ, Roethling F, Haschke R, Ritter H (2004) Situated robot learning for multi-modal instruction and imitation of grasping. Robotics and Autonomous Systems Special Issue(47):129–141
Stoytchev A (2003) Computational Model for an Extendable Robot Body Schema. Tech. Rep. GIT-CC-03-44, Georgia Institute of Technology, College of Computing
Sugiura K, Iwahashi N, Kashioka H, Nakamura S (2010) Statistical Imitation Learning in Sequential Object Manipulation Tasks. Advances in Robot Manipulators pp 589–606
Toussaint M, Gienger M, Goerick C (2007) Optimization of sequential attractor-based movement for compact behaviour generation. In: 7th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2007)
Toussaint M, Plath N, Lang T, Jetchev N (2010) Integrated motor control, planning, grasping and high-level reasoning in a blocks world using probabilistic inference. In: Robotics and Automation (ICRA), 2010 IEEE International Conference on, IEEE, pp 385–391
Walther D, Koch C (2006) Modeling attention to salient proto-objects. Neural Networks 19(9):1395–1407
Wischnewski M, Belardinelli A, Schneider WX, Steil JJ (2010) Where to Look Next? Combining Static and Dynamic Proto-objects in a TVA-based Model of Visual Attention. Cognitive Computation pp 1–18
Yamashita Y, Tani J (2008) Emergence of Functional Hierarchy in a Multiple Timescale Neural Network Model: A Humanoid Robot Experiment. PLoS computational biology 4(11)